



SPECTEROPS



TROOPERS

# Misconfiguration Manager

Overlooked and Overprivileged

Chris Thompson & Duane Michael  
SpecterOps



Scan to download this slide deck

# Duane Michael

- Managing Consultant, Adversary Simulation at SpecterOps
- Contributor to SharpSCCM
- @subat0mik on all the things

# Chris Thompson

- Principal Consultant, Adversary Simulation at SpecterOps
- Primary author of SharpSCCM
- X: @\_Mayyhem

# Garrett Foster

- Senior Consultant, Adversary Simulation at SpecterOps
- Primary author of SCCMHunter
- X: @garrfoster





**Enough about us.  
Let's get to know you!**

# Agenda

What this talk is (and is not) about



## This presentation covers:

- Brief SCCM introduction
- Exposure to common SCCM attack paths
- Hierarchy takeover demo
- Stories from the field
- Intro to our SCCM attack path management project



## This presentation does NOT cover:

- Walkthroughs of *all* offensive techniques
- Specific defensive walkthroughs
- Comprehensive treatment of topics discussed

# SCCM Introduction

## Laying the groundwork

- What is Microsoft Configuration Manager?
  - Previously System Center Configuration Manager (SCCM)
  - Used for wide-scale deployment of applications, software updates, operating systems, and compliance settings
  - Real-time management of servers, desktops, and laptops

# SCCM Introduction

## Know your target

- As an attacker, why should I care?
  - Used by many organizations that use Windows workstations, so you're likely to encounter it
  - Often used to manage clients in multiple AD domains/forests and networks, crossing trust boundaries and bypassing segmentation
  - Commonly misconfigured due to some insecure default settings and poor community advice

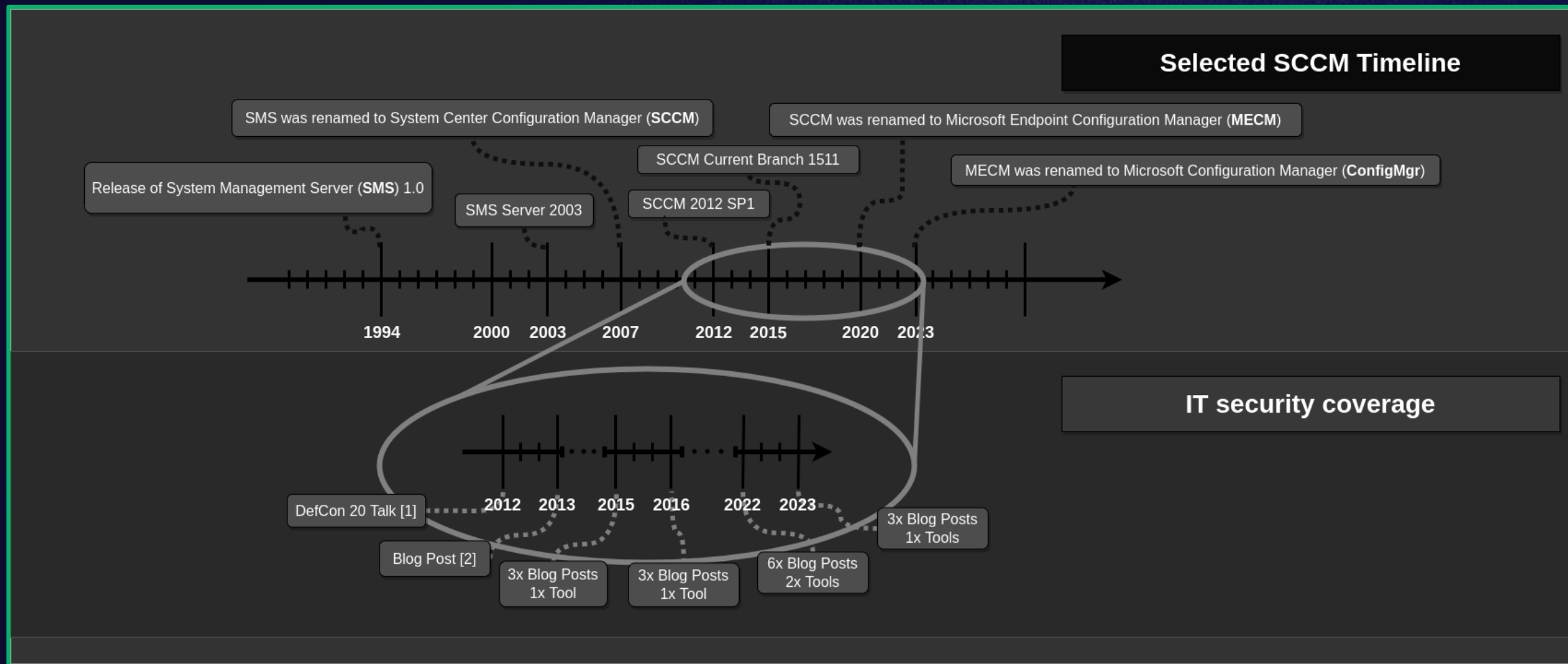


# SCCM Introduction

Know your attack surface

- As a defender or administrator, why should I care?
  - If you work in a Windows/Active Directory enterprise environment, you're likely using SCCM
  - Misconfigurations cause dangerous vulnerabilities that may lead to domain compromise

# A Brief History of SCCM Security Research





# SCCM Primer

## The building blocks

### ***Hierarchy***

- One instance of SCCM, consisting of one or more sites
- The security boundary in SCCM

### ***Site***

- An environment that provides services to a scope of clients
- Identified by a three-character site code (e.g., PS1)

### ***Client/Device***

- Systems joined to, managed by, and that receive content from an SCCM site through installation of the SCCM client software (think C2 agent)

# SCCM Primer

## The building blocks

### ***Primary Site***

- A site that clients can be assigned to
- Administered using the Configuration Manager console

### ***Primary Site Server***

- Handles processing of client data in a primary site
- Also referred to as just the “site server”

### ***Site Database Server***

- The server(s) that host the database where client and server data is stored for the primary site

# SCCM Primer

## Hierarchy structure

### ***Central Administration Site (Optional)***

- Used to manage 2+ primary sites
- Only needed for >150k clients

### ***Standalone Primary Site***

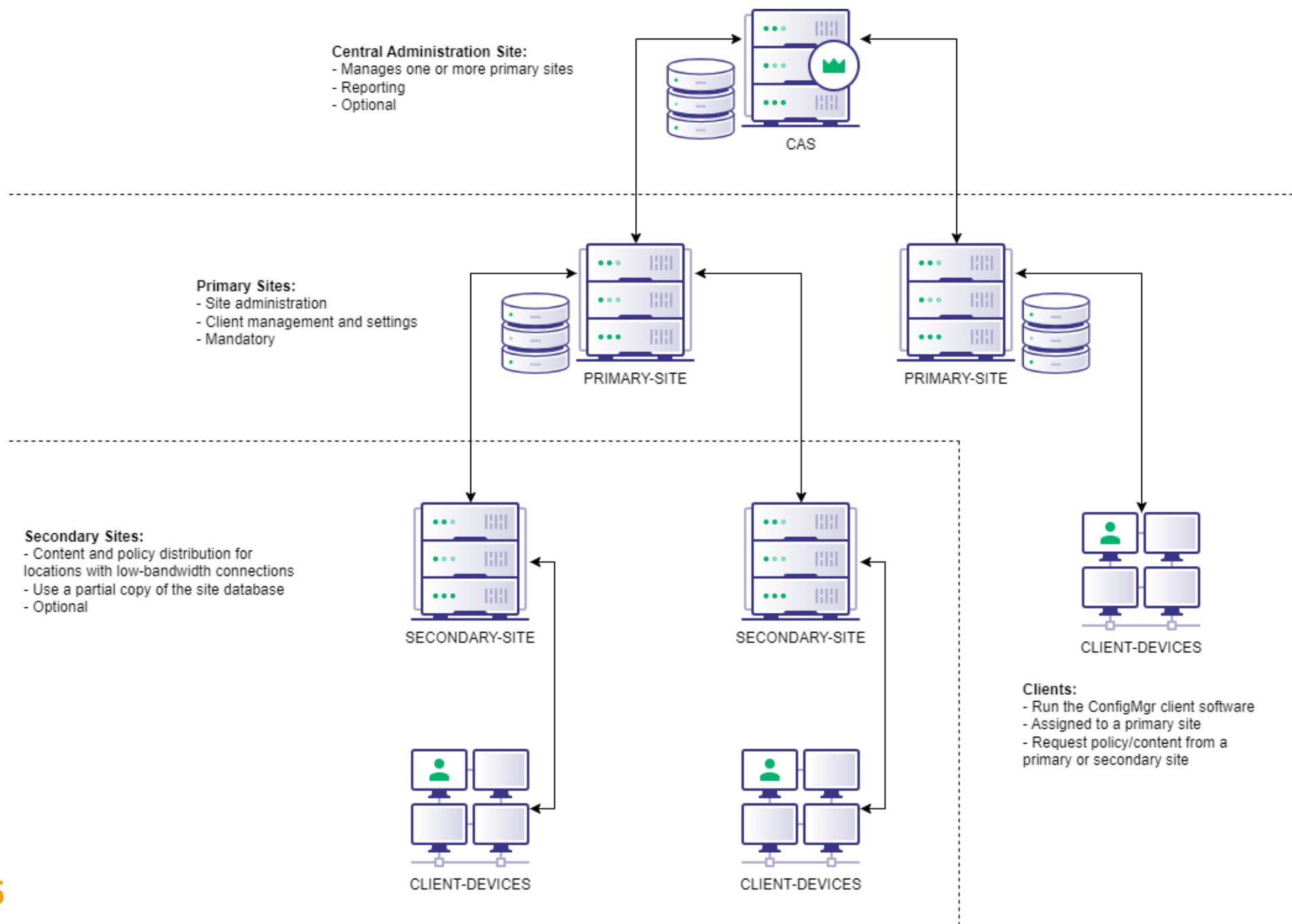
- One site to rule them all, recommended by Microsoft

### ***Secondary Sites (Optional)***

- Primary sites can contain multiple secondary sites
- Allow software deployment to remote locations with limited bandwidth







# SCCM Primer

## Site system roles

### ***SMS Provider***

- Interface for the console to interact with the site database via WMI or REST API
- Allows indirect access to the site database
- Installed on the primary site server by default but can also be installed elsewhere

### ***Configuration Manager console***

- The software that administrators use to manage a site via an SMS Provider

# SCCM Primer

## Site system roles

### ***Management Point:***

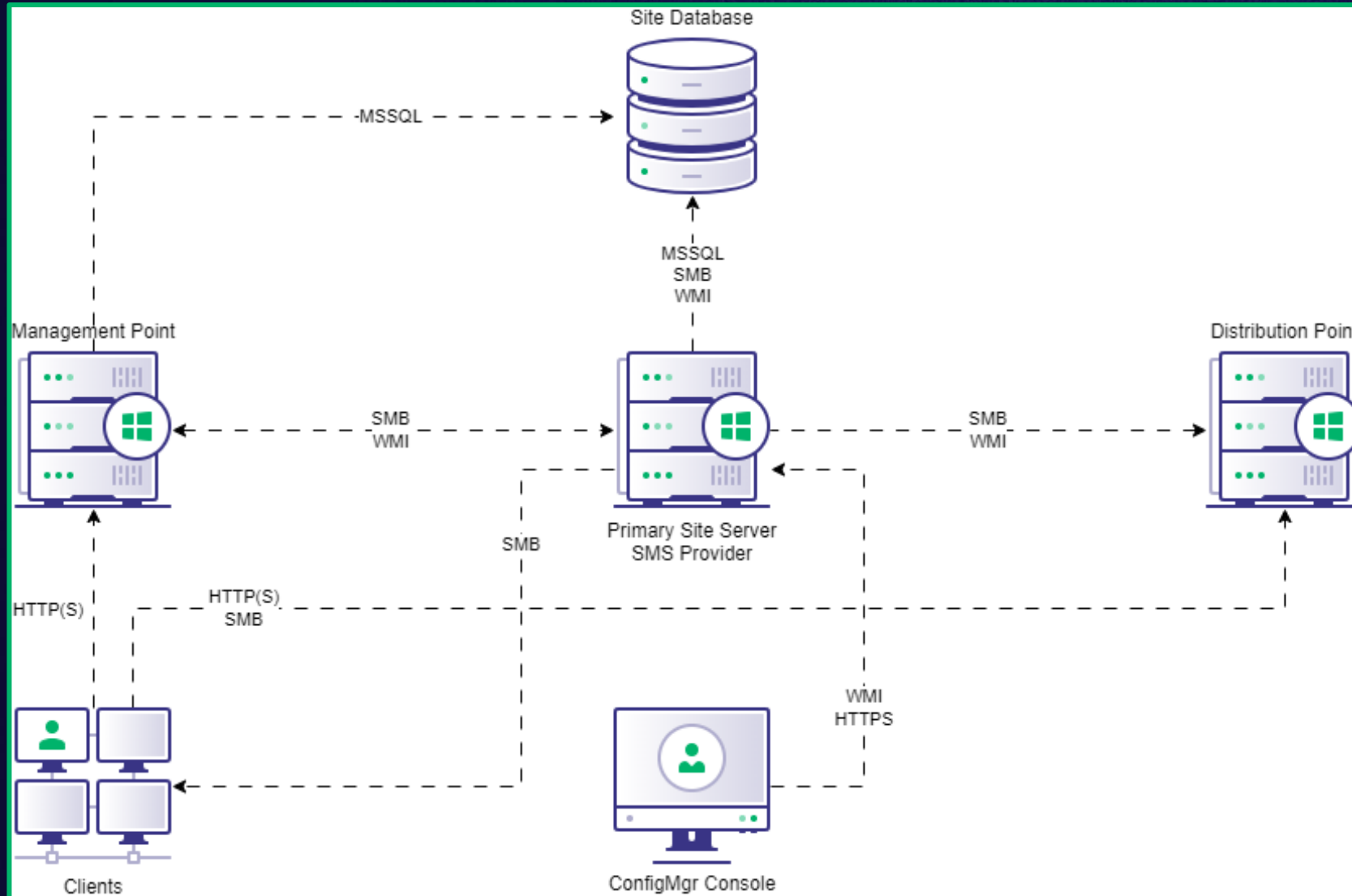
- Receives client status and inventory messages via HTTP(S)
- Responds to client requests for policy and content locations

### ***Distribution Point:***

- Receives and responds to client requests for content via HTTP(S)/SMB
  - Applications, software packages, scripts, etc.
- Clients download software from distribution points



# Site Communication Protocols



# SCCM has *many* accounts...

Many accounts are used for many things, most are abusable...



## Client Push Installation

- Used to install the client software on computers
- Must be admin on every target computer
- Results in many overprivileged scenarios



## Network Access

- Used to retrieve software from DPs
- (Sometimes) optional but still wide-spread
- Stored on clients (DPAPI) and transmitted via computer policy (obfuscated, not encrypted)



## Task Sequence

Various accounts:

- Domain join account
- RunAs account
- Network folder connection account
- Collection variables

# Client Push Installation

## How computers become clients

- Used to deploy the SCCM client software remotely from the site server
- Copies installation files to the ADMIN\$ share and executes ccmsetup.exe
- Uses configured accounts and the site server domain computer account, which must be a local admin to successfully install or reinstall the client software





# Automatic site-wide client push installation

## How computers become clients

- The site server automatically tries client push installation on any computers it discovers in the domain or network
- Can be abused by creating fake device records, which cause the site server to connect to the ADMIN\$ share at an arbitrary IP address
- Incoming NTLM authentication to the IP address can be relayed to other workstations or SCCM servers (where the site server has admin privileges)



# Network Access Accounts

What are they and why do they exist?

- Domain account used to retrieve software from distribution points (DP)
- (Mostly) optional, required for specific actions / scenarios
- Requires minimal privileges: read the network share on the DP



# The Worst (and Most Common) Misconfiguration

## Overprivileged Network Access Accounts

- Included in computer policy sent to all clients
- Policy can be requested with control of a computer object
- Credentials are obfuscated on the wire (no encryption)
- Protected by DPAPI on the client, recoverable as admin





# The Worst (and Most Common) Misconfiguration

## Overprivileged Network Access Accounts

- Due to so many different accounts, the same god-mode account is often used
  - e.g., Domain Admin, SCCM admin, client push installation (local admin on all clients)
- **We find this *All. The. Time.***
- Creds may persist beyond account rotation





```
PS C:\Users\labadmin.APERTURE\Desktop> .\SharpSCCM.exe local secrets -m wmi
```



🔍 Search



10:29 PM  
6/11/2024



# Hierarchy Takeover

Assuming full control of all systems in the SCCM hierarchy

## Why do we care?

- Allows arbitrary *command execution* on all clients
- Allows access to features like CMPivot, Run Script
- Allows ability to impact availability of software



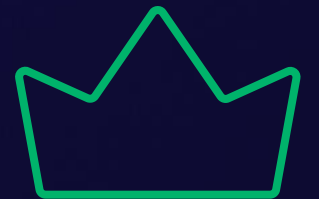


# Hierarchy Takeover

Assuming full control of all systems in the SCCM hierarchy

## How can attackers take over a hierarchy?

- Obtain the **Full Administrator** role in **ANY** site
- There is ***no security boundary*** between sites in the same hierarchy
- The site database is replicated to all sites (e.g., admin users)
- Own one primary site, ***own them all***



# NTLM Relay Primer

## Connecting the dots

If an account authenticates to an attacker-controlled machine via NTLM, the attacker can forward the authentication to another system to access it using the relayed account's privileges

- e.g., to launch a C2 agent, add a user account, modify permissions/configurations, etc.

Several bugs that Microsoft won't fix can be abused to force a computer to authenticate to an arbitrary IP address using NTLM (a.k.a. coercion)

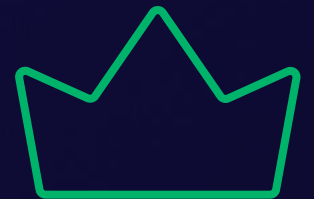
- Printerbug
- PetitPotam

# Hierarchy Takeover

## Key concepts

- The primary site server's computer account **must** be:
  - Local admin on the site database server
  - Sysadmin on the site database
  - Local admin on every other site system role

If we can **coerce authentication from this account** and relay the authentication to certain SCCM servers, we **gain control of SCCM.**

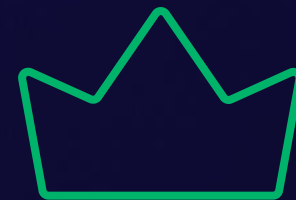




# Hierarchy Takeover Attack Paths

Just a few examples...

- Coerce NTLM from site server or SMS Provider → Relay to MSSQL on remote site DB  
→ Grant Full Admin
- Coerce NTLM from site server → Relay to SMB on remote site DB server  
→ Compromise/impersonate DBA, Grant Full Admin
- Coerce NTLM from site server → Relay to HTTPS on remote SMS Provider  
→ Grant Full Admin
- Coerce NTLM from site server → Relay to SMB on remote SMS Provider server  
→ Grant Full Admin via WMI
- *And many, many more...*



Site Server



Attacker Host



Site Database



Site Server



Attacker Host



Site Database



Coerce NTLM authentication





Site Server



Attacker Host



Site Database



Coerce NTLM authentication

Connect to smb://<attacker\_ip>/<path>

Site Server



Attacker Host



Site Database



Coerce NTLM authentication

Connect to smb://<attacker\_ip>/<path>

Connect to mssql://<database\_ip>/<path>

Site Server



Attacker Host



Site Database



Coerce NTLM authentication

Connect to smb://<attacker\_ip>/<path>

Connect to mssql://<database\_ip>/<path>

NTLM challenge



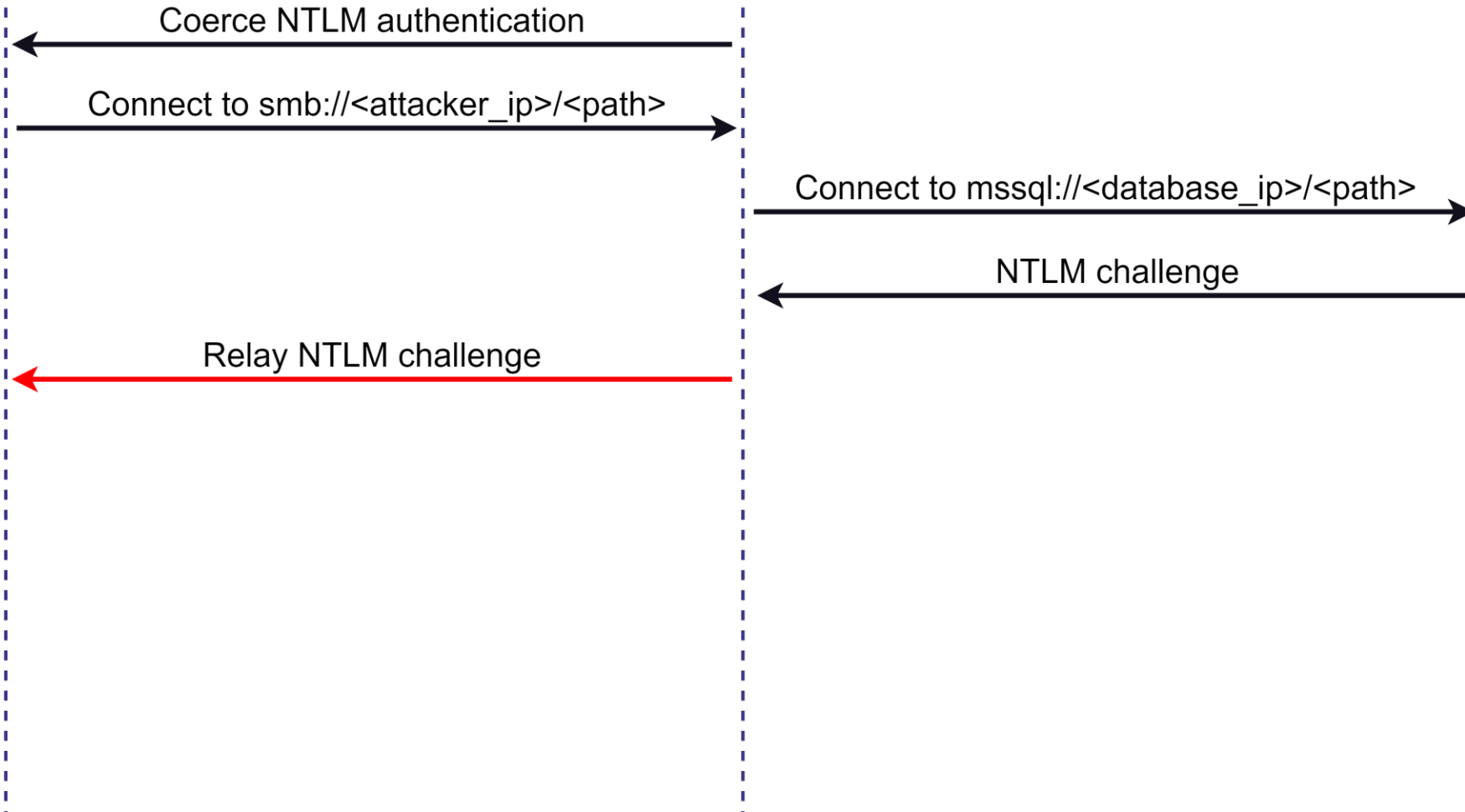
Site Server



Attacker Host



Site Database



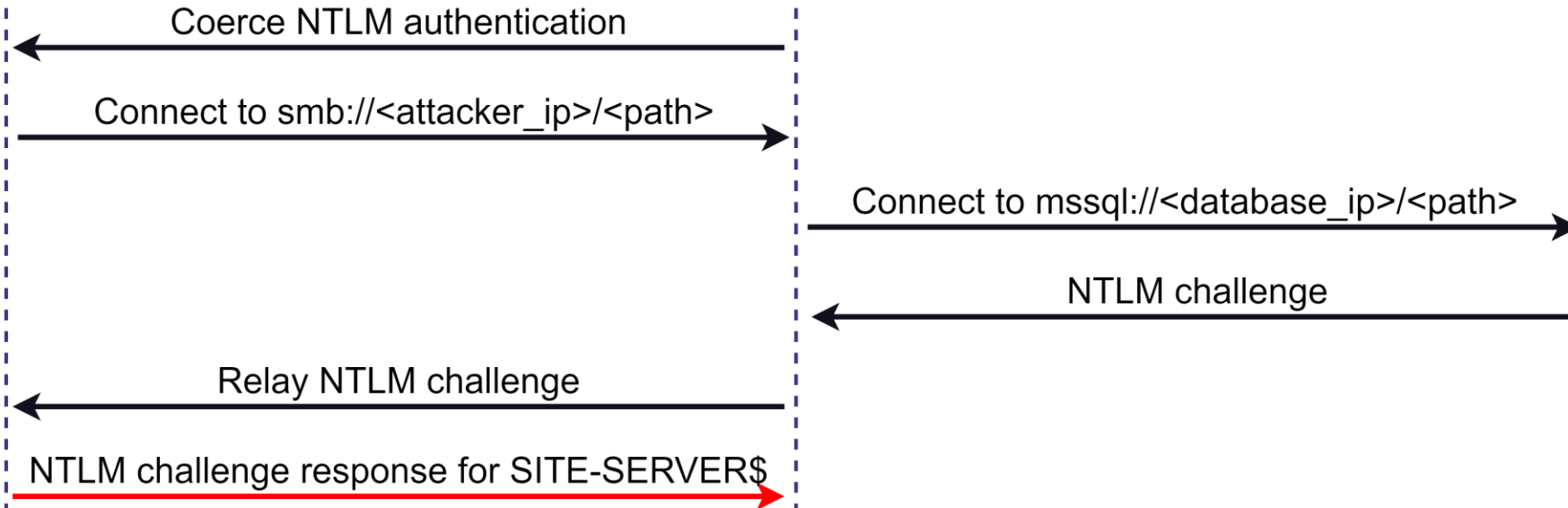
Site Server



Attacker Host



Site Database



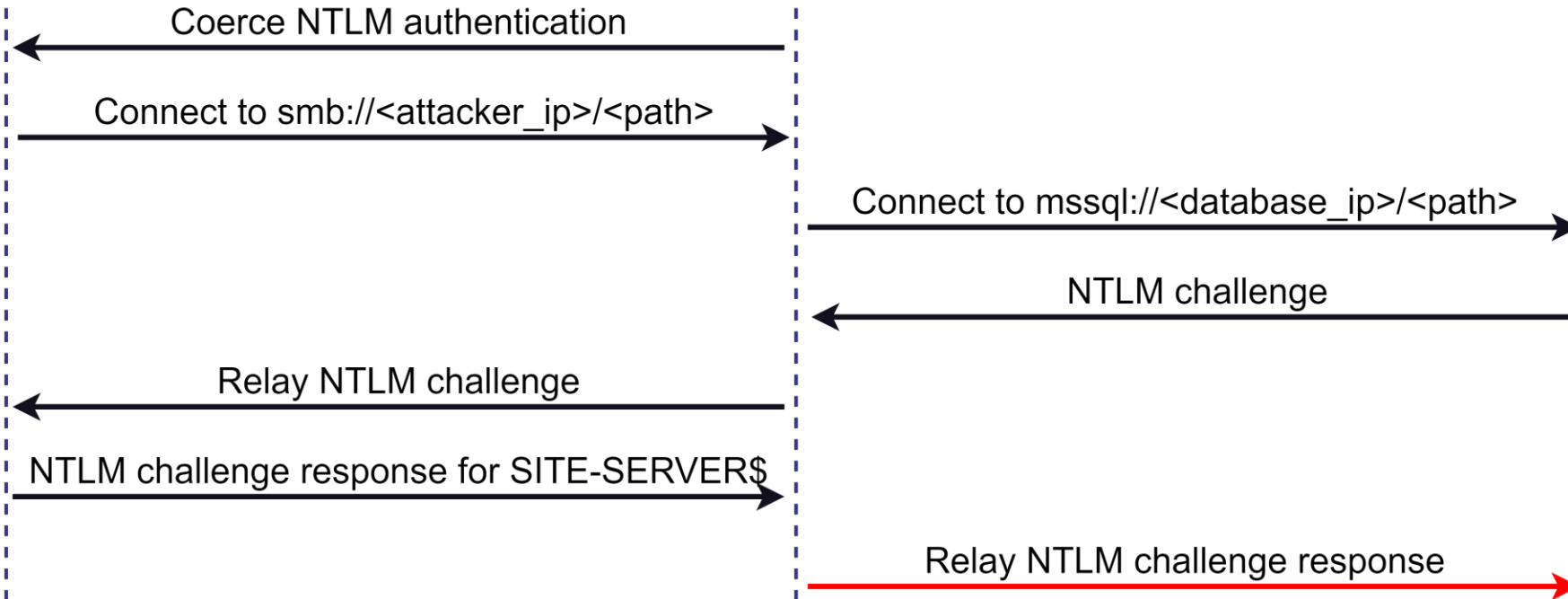
Site Server



Attacker Host



Site Database





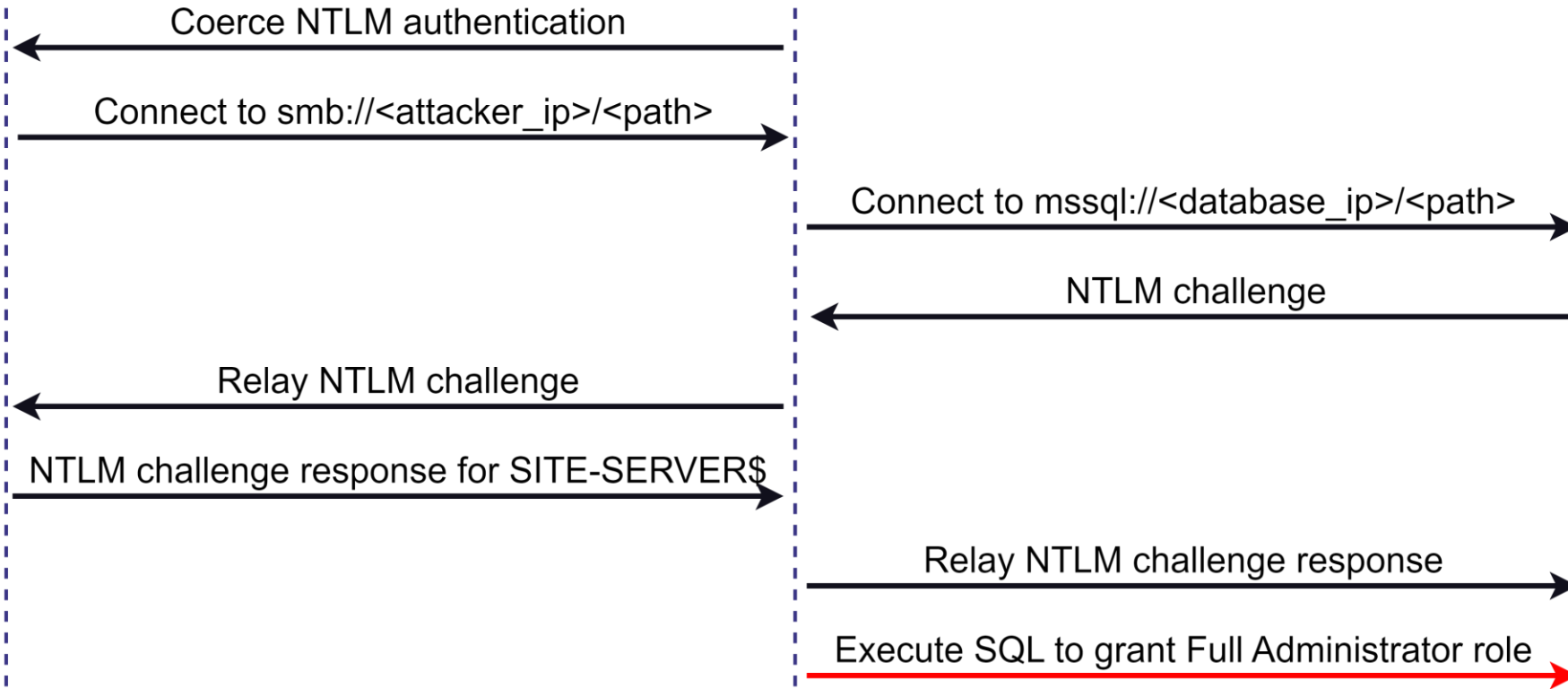
Site Server



Attacker Host



Site Database





Home

Add User  
or Group  
CreateSaved  
Searches  
Search

Administration Overview Security Administrative Users

## Administration

## Overview

- Updates and Servicing
- Hierarchy Configuration
- Cloud Services
- Site Configuration
- Client Settings
- Security
  - Administrative Users

Assets and Compliance

Software Library

Monitoring

Administration

Community

## Administrative Users 2 items

Search current node



Search

Add Criteria

Icon	Account Name	Account Display Name	Security Roles
	APERTURE\labadmin		"Full Administrator"
	SITE-SERVER\labadmin		"Full Administrator"

Ready



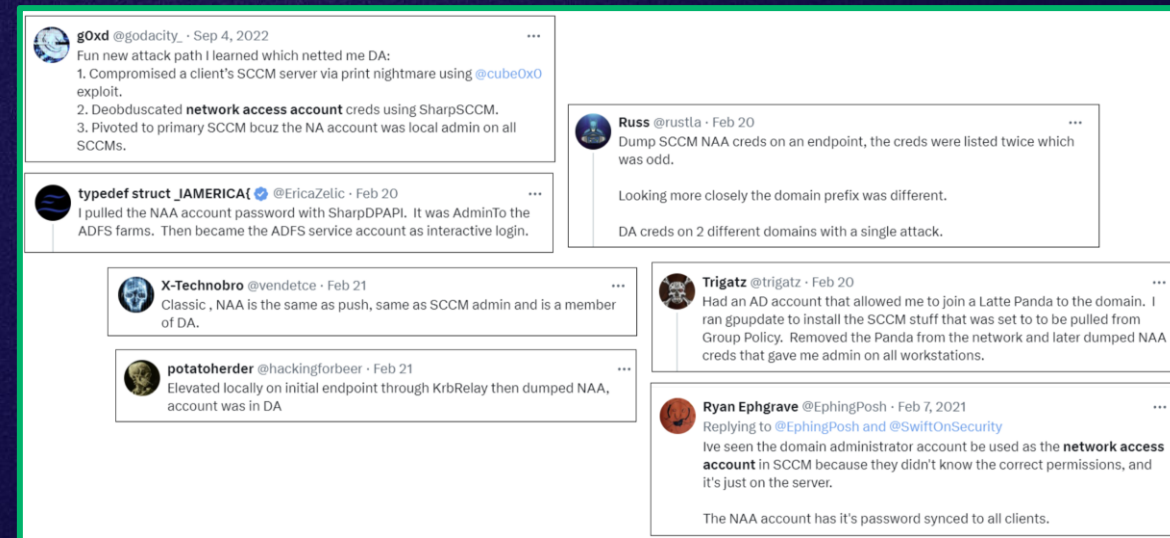
Type here to search

7:21 PM  
5/4/2024

# The Perils of Excess: A Tale of Unbridled Access and Forgotten Accounts in SCCM

## Overprivileged Network Access Accounts

1. Local admin (LA) on every client found in PXE media on SharePoint
2. Configured with client push installation account (LA everywhere)
3. Configured with DA account
4. LA on every SCCM site server
5. Previous (legacy) NAAs recovered from CIM Repository
6. Two DA accounts (disparate domains) configured - @rustla



# Booting Up to Boss Level: A Domain Controller's Unexpected Journey

## When domain join accounts own the domain

- SCCM domain-join accounts (DJA) are used to join new computers to the domain after PXE booting
- Pushed out via task sequence policy
- The account used to join a computer has ownership rights on the computer
- This account joined servers to the domain which were later promoted DCs
- DJA (present on all clients) had ownership rights over DCs





# Why not both?

~~Domain Controllers~~ SCCM Clients

- Sites can be configured to enroll domain controllers as clients
- If we can takeover the site, we can compromise the domain controller through remote execution
- SCCM execution methods:
  - Application deployment
  - Script deployment
  - Package deployment



# Crawling Through the Darkness

From Random Connection String to SCCM Admin

1. Connection string found in script on network share
2. Authenticate to the MSSQL DB
3. Crawl three SQL links, last has DBA in the SCCM site DB
4. Dump/crack DBA credentials
5. Connect to SCCM site DB
6. Grant Full Admin
7. Host C2 payload on public file share
8. Execute beacons on client domain controllers as SYSTEM





# Love at First Site: The Unyielding Pursuit of a Laptop Long Gone

## Client Push Installation to Non-existent Machines

1. Automatic client push installation attempted to authenticate to computers that no longer existed
2. Site server attempted to authenticate to the CISO's old laptop.. every hour... for two years...
3. Create an ADIDNS record for the computer name, point it at our machine, capture/relay the authentication







Now that you see what's possible...



# Misconfiguration Manager

## Helping you manage SCCM attack paths

- Living knowledge-base that aims to ease SCCM attack path management
- Contains foundational, offensive, and defensive write-ups for most known techniques
- Introduces a taxonomy to simplify and demystify concepts (à la Certified Pre-Owned)
- Based on MITRE ATT&CK and inspired by the SaaS Attacks Matrix

<https://github.com/pushsecurity/saas-attacks>  
<https://attack.mitre.org/>

# Misconfiguration Manager

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration
PXE Credentials	App Deployment	App Deployment	Relay to Site Server SMB	App Deployment	PXE Credentials	LDAP Enumeration	App Deployment	CMPivot		CMPivot
	Script Deployment	Script Deployment	Relay Client Push Installation	Script Deployment	Policy Request Credentials	SMB Enumeration	Script Deployment			
		ADCS Relay	Relay to DB MSSQL		DPAPI Credentials	HTTP Enumeration	Relay to Site Server SMB			
		LDAP Relay	Relay to DB SMB		Legacy Credentials	CMPivot	Relay Client Push Installation			
			Relay to ADCS				Relay to DB MSSQL			
			Relay to AdminService		Site Database Credentials		Relay to DB SMB			
			Relay CAS to Child				Relay CAS to Child			
			Relay to SMS Provider SMB				Relay to AdminService			
			Relay between HA				Relay to SMS Provider SMB			

# Misconfiguration Manager Taxonomy

Because "Hierarchy takeover via NTLM coercion and relay to MSSQL on remote site database" does not roll off the tongue...



## CRED

1. Retrieve credentials from PXE boot media
2. Deobfuscate computer policy
3. Decrypt via DPAPI
4. Legacy credentials (DPAPI)
5. SC\_UserAccount on Site DB



## ELEVATE

1. SMB relay on site server
2. Automatic client push NTLM relay



## EXEC

1. Application deployment
2. Script deployment



## RECON

1. LDAP Enumeration
2. SMB Enumeration
3. HTTP(S) Enumeration
4. CMPivot



# SCCM Hierarchy Takeover Attack Paths

Because "Hierarchy takeover via NTLM coercion and relay to MSSQL on remote site database" does not roll off the tongue...



## TAKEOVER-1

NTLM coercion and relay to  
MSSQL on remote site database



## TAKEOVER-2

NTLM coercion and relay to SMB  
on remote site database



## TAKEOVER-3

NTLM coercion and relay to HTTP  
on ADCS



## TAKEOVER-4

NTLM coercion and relay from  
CAS to origin primary site server



## TAKEOVER-5

NTLM coercion and relay to  
AdminService on remote SMS Provider



## TAKEOVER-6

NTLM coercion and relay to SMB  
on remote SMS Provider



## TAKEOVER-7

NTLM coercion and relay to SMB between  
primary and passive site servers



## TAKEOVER-8

NTLM coercion and relay HTTP to  
LDAP on domain controller



# SCCM Mitigation and Detection Guidance

You didn't think we'd leave you hanging, did you?



## PREVENT

Currently 23 SCCM and AD configuration changes to mitigate the attack techniques covered



## DETECT

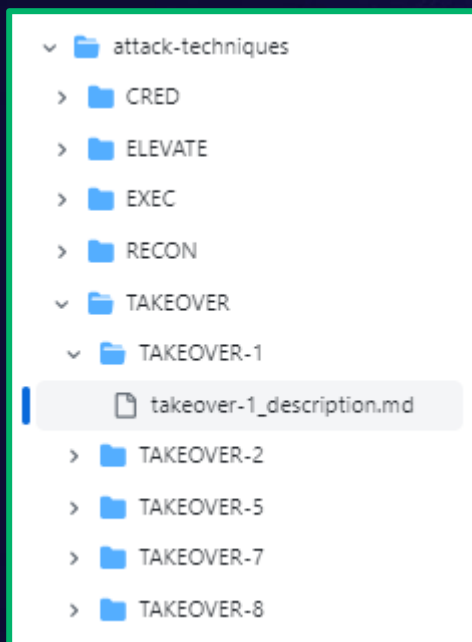
Strategies to detect SCCM attack techniques and attack paths



## CANARY

Deception techniques that take advantage of SCCM misconfigurations

# Misconfiguration Manager: A Glimpse



## TAKEOVER-1

### Description

Hierarchy takeover via NTLM coercion and relay to MSSQL on remote site database

### MITRE ATT&CK TTPs

- [TA0008](#) - Lateral Movement
- [TA0004](#) - Privilege Escalation

### Requirements

#### Coercion

- Valid Active Directory domain credentials
- Connectivity to SMB (TCP/445) on a coercion target:
  - TAKEOVER-1.1: Coerce primary site server
  - TAKEOVER-1.2: Coerce SMS Provider
  - TAKEOVER-1.3: Coerce passive site server
- Connectivity from the coercion target to SMB (TCP/445) on the relay server
- Coercion target settings:
  - `BlockNTLM` = `0` or not present, or = `1` and `BlockNTLMServerExceptionList` contains attacker relay server
  - `RestrictSendingNTLMTraffic` = `0`, `1`, or not present, or = `2` and `ClientAllowedNTLMServers` contains attacker relay server
  - Domain computer account is not in `Protected Users`
- Domain controller settings:
  - `RestrictNTLMInDomain` = `0` or not present, or is configured with any value and `DCAllowedNTLMServers` contains coercion target
  - `LmCompatibilityLevel` < `5` or not present, or = `5` and `LmCompatibilityLevel` >= `3` on the coercion target

#### Relay

- Connectivity from the relay server to MSSQL (TCP/1433) on the relay target, the site database
- Extended protection for authentication not required on the site database
- Relay target settings:
  - `RequireSecuritySignature` = `0` or not present
  - `RestrictReceivingNTLMTraffic` = `0` or not present
  - Coercion target is local admin (to access RPC/admin shares)
- Domain controller settings:
  - `RestrictNTLMInDomain` = `0` or not present, or is configured with any value and `DCAllowedNTLMServers` contains relay target

## Summary

By default, the Active Directory domain computer accounts for primary site servers, systems hosting the SMS Provider role, CAS site servers, and passive site servers are granted the `db_owner` role in their respective site's MSSQL database. An attacker who is able to successfully coerce NTLM authentication from one of these accounts and relay it to the site database can use these permissions to grant an arbitrary domain account the SCCM "Full Administrator" role.

## Impact

The "Full Administrator" security role is granted all permissions in Configuration Manager for all scopes and all collections. An attacker with this privilege can execute arbitrary programs on any client device that is online as SYSTEM, the currently logged on user, or as a specific user when they next log on. They can also leverage tools such as CMPivot and Run Script to query or execute scripts on client devices in real-time using the AdminService or WMI on an SMS Provider.

## Defensive IDs

- [PREVENT-2: Disable Fallback to NTLM](#)
- [PREVENT-12: Require SMB signing on site systems](#)
- [PREVENT-14: Require Extended Protection for Authentication \(EPA\) on AD CS CAs and standalone site databases](#)

## Subtechniques

- TAKEOVER-1.1: Coerce primary site server
- TAKEOVER-1.2: Coerce SMS Provider
- TAKEOVER-1.3: Coerce passive site server

## Examples

The steps to execute TAKEOVER-1.1 through TAKEOVER-1.3 are the same except that a different system is targeted for coercion of NTLM authentication.

1. (Linux) Use `sccmhunter` to get the hex-formatted SID of the Active Directory user you'd like to grant the Full Administrator role in SCCM, as well as the MSSQL statements required to grant the role to the user:

```
$ python3 sccmhunter.py mssql -dc-ip 192.168.57.100 -d MAYYHEM.LOCAL -u 'lowpriv' -p 'P@ssw0rd' -debug -tu lowpriv -sc ps1

[13:13:33] DEBUG    [+] Bind successful ldap://192.168.57.100:389 - cleartext
[13:13:33] INFO     [*] Resolving lowpriv SID...
[13:13:33] DEBUG    [+] Found lowpriv SID: S-1-5-21-622943703-4251214699-2177406285-1112
[13:13:33] INFO     [*] Converted lowpriv SID to 0x010500000000000515000000D75D21256B6364FD4D95C88158040000
[13:13:33] DEBUG    [+] Found domain netbiosname: MAYYHEM
[13:13:33] INFO     [*] Use the following to add lowpriv as a Site Server Admin.

USE CM_ps1; INSERT INTO RBAC_Admins (AdminSID,LogonName,IsGroup,IsDeleted,CreatedBy,CreatedDate,ModifiedBy,ModifiedDate,Sou
```

(Windows) Use `sharpSCCM` to get the hex-formatted SID of the Active Directory user you'd like to grant the Full Administrator role in SCCM, and assemble the query based on the output from the example `sccmhunter` command above, substituting the user SID, domain, and site code ( `ps1` in this example) where appropriate.

On Windows, using `SharpSCCM` :

```
> .\SharpSCCM.exe get users -n lowpriv -sms SITE-SMS -sc ps1

[+] Connecting to \\SITE-SMS\root\SMS\site_ps1
[+] Executing WQL query: SELECT * FROM SMS_R_User WHERE UniqueUserName LIKE '%lowpriv%'
-----
SMS_R_User
-----
AADTenantID:
AADUserID:
ADObjectCreationTime: 20230721132400.000000+***
AgentName: SMS_AD_USER_DISCOVERY_AGENT, SMS_AD_SECURITY_GROUP_DISCOVERY_AGENT
AgentSite: PS1, PS1
AgentTime: 20230721202501.000000+***, 20230803202502.000000+***
CloudUserId:
CreationDate: 20230721202502.760000+***
DistinguishedName: CN=Low Priv,CN=Users,DC=MAYYHEM,DC=LOCAL
FullDomainName: MAYYHEM.LOCAL
FullUserName: Low Priv
Mail:
Name: MAYYHEM\lowpriv (Low Priv)
NetworkOperatingSystem: Windows NT
ObjectGUID: Can't display UInts as a String
PrimaryGroupID: 513
ResourceId: 2063597571
ResourceType: 4
SecurityGroupName: MAYYHEM\Domain Users
SID: S-1-5-21-622943703-4251214699-2177406285-1112
UniqueUserName: MAYYHEM\lowpriv
UserAccountControl: 66048
UserContainerName: MAYYHEM\USERS
UserGroupName: MAYYHEM\Domain Users
UserName: lowpriv
UserOUName:
UserPrincipalName: lowpriv@MAYYHEM.LOCAL
WindowsNTDomain: MAYYHEM
-----
[+] Completed execution in 00:00:00.9878140
```

## References

- Chris Thompson, SCCM Site Takeover via Automatic Client Push Installation, <https://posts.specterops.io/sccm-site-takeover-via-automatic-client-push-installation-f567ec80d5b1>
- Chris Thompson, SCCM Hierarchy Takeover: One Site to Rule Them All, <https://posts.specterops.io/sccm-hierarchy-takeover-41929c61e087>
- Garrett Foster, SCCM Hierarchy Takeover with High Availability, <https://posts.specterops.io/sccm-hierarchy-takeover-with-high-availability-7dcbd3696b43>
- Garrett Foster, sccmhunter, <https://github.com/garrettfoster13/sccmhunter>
- Chris Thompson, SharpSCCM, <https://github.com/Mayhem/SharpSCCM>

## PREVENT-4

### Description

Configure Enhanced HTTP

### Summary

[Enhanced HTTP](#) (eHTTP) is a simplified method of secure communication without the overhead of a standard PKI deployment. In an eHTTP setup, the site issues self-signed certificates to the various site servers, such as management points and distribution points. Then, these site systems issue unique site tokens to clients. The client then uses the site token for communication with site servers. Microsoft provides a diagram of this process (Figure 1).

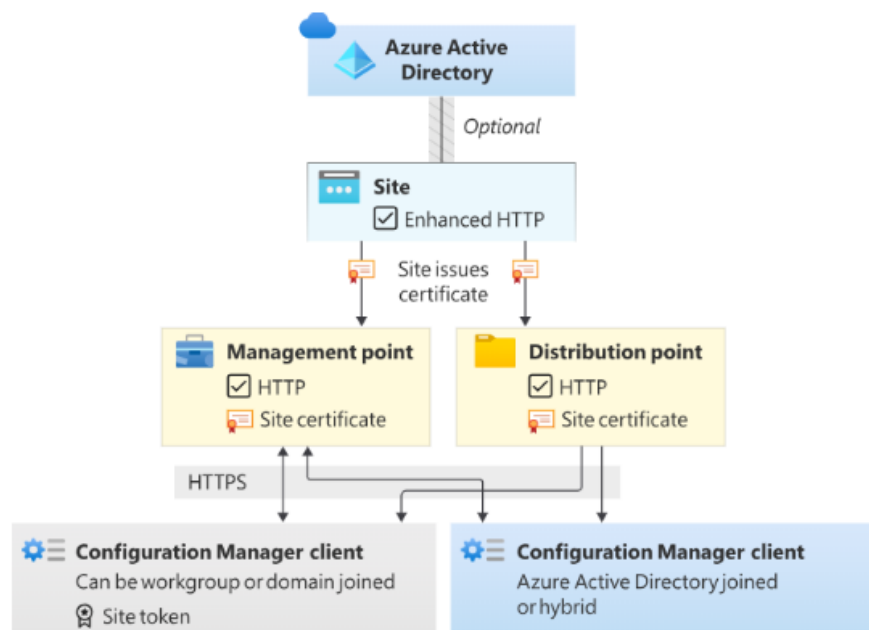


Figure 1 - Enhanced HTTP Diagram

**NOTE:** The preferred/recommended method for secure communication is the use of PKI certificates ([PREVENT-8](#)). eHTTP is a compromise between PKI and standard HTTP use and is certainly a better option than the latter.

### Linked Defensive IDs

- [PREVENT-3: Harden or Disable Network Access Account](#)
- [PREVENT-8: Require PKI certificates for client authentication](#)
- [PREVENT-15: Disable legacy network access accounts in Active Directory](#)

### Associated Offensive IDs

- [CRED-2: Request and deobfuscate machine policy to retrieve credential material](#)
- [CRED-3: Dump network access account \(NAA\) credentials via WMI](#)
- [CRED-4: Retrieve legacy network access account \(NAA\) credentials from the CIM Repository](#)

### References

- Christopher Panayi, An inside look: How to distribute credentials securely in SCCM, <https://www.mwrcybersec.com/an-inside-look-how-to-distribute-credentials-securely-in-sccm>
- Microsoft, Enhanced HTTP, <https://learn.microsoft.com/en-us/mem/configmgr/core/plan-design/hierarchy/enhanced-http>



```
PS C:\Users\labadmin\Downloads> .\MisconfigurationManager.ps1 -Verbose
```

```
VERBOSE: Looking for site namespace in root\SMS on SITE-SERVER
```

```
VERBOSE: Found root\SMS\site_PS1 on SITE-SERVER
```

```
VERBOSE: Querying root\SMS\site_PS1.SMS_SCI_SiteDefinition for the list of sites with parent:
```

```
■
```

# Misconfiguration Manager

## Future Work

- Microsoft is collaborating and taking this seriously!
- There is SO much more work to be done:
  - Offensive research
  - Detection strategies
  - Configuration guidance
- We want to hear your stories and ideas!
- Pull requests welcome and encouraged
- Collaborate with us in [#sccm](#) on BloodHound Slack
  - Invite link: <https://ghst.ly/BHSlack>





SPECTEROPS



TROOPERS

# Thank you!

Chris Thompson | [@\\_Mayyhem](#)

Duane Michael | [@subat0mik](#)

Garrett Foster | [@garrfoster](#)



Scan to download this slide deck





SPECTEROPS



TROOPERS

# Questions?

Chris Thompson | [@\\_Mayyhem](#)

Duane Michael | [@subat0mik](#)

Garrett Foster | [@garrfoster](#)



Scan to download this slide deck