Immunity

Origin Information Disclosure

2020-07-29

Table of Contents

Advisory Information

Title: Origin Information Disclosure
Vendors contacted: Electronic Arts Inc.
Release mode: Coordinated Release
Credits: This vulnerability was discovered by Andres Blanco.

Vulnerability Information

Class: Insertion of Sensitive Information Into Debugging Code [CWE-215]
Affected Version: Origin 10.5.72.41482 (prior versions might be affected)
Remotely Exploitable: No
Locally Exploitable: Yes
Severity: Medium - 5.5 (CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N 5.5)
CVE Identifier: CVE-2020-13172

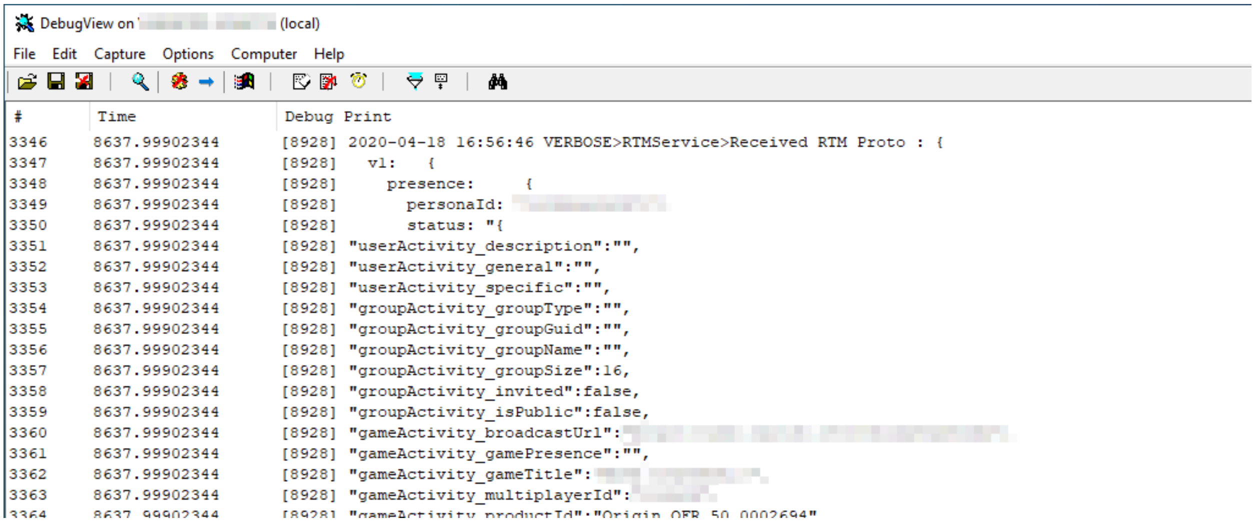
Vulnerability Description

Origin is a digital distribution platform developed by Electronic Arts for purchasing and playing video games.

This application uses the OutputDebugString[1] function to display sensitive user information. The mentioned API function from KERNEL32.DLL lets programs send messages to a debugger and it's important to note that any process on the system, including a non-privileged one, can process these messages[2].

[1] https://docs.microsoft.com/en-us/windows/win32/api/debugapi/nf-debugapi-outputdebugstringa

[2] http://unixwiz.net/techtips/outputdebugstring.html

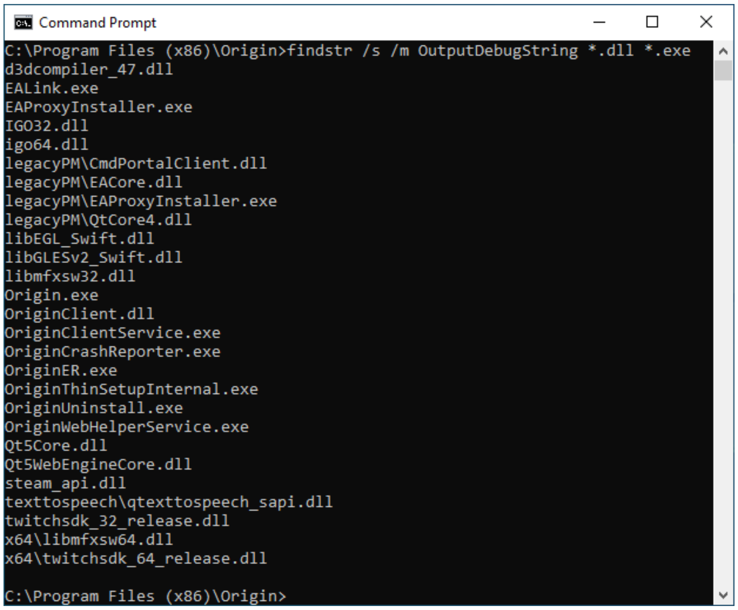


DebugView on [redacted] (local)
File Edit Capture Options Computer Help

```
#          Time                 Debug Print
3346       8637.99902344        [8928] 2020-04-18 16:56:46 VERBOSE>RTMService>Received RTM Proto : {
3347       8637.99902344        [8928]   v1:   {
3348       8637.99902344        [8928]     presence:     {
3349       8637.99902344        [8928]       personaId: [redacted]
3350       8637.99902344        [8928]       status: "{
3351       8637.99902344        [8928] "userActivity_description":"",
3352       8637.99902344        [8928] "userActivity_general":"",
3353       8637.99902344        [8928] "userActivity_specific":"",
3354       8637.99902344        [8928] "groupActivity_groupType":"",
3355       8637.99902344        [8928] "groupActivity_groupGuid":"",
3356       8637.99902344        [8928] "groupActivity_groupName":"",
3357       8637.99902344        [8928] "groupActivity_groupSize":16,
3358       8637.99902344        [8928] "groupActivity_invited":false,
3359       8637.99902344        [8928] "groupActivity_isPublic":false,
3360       8637.99902344        [8928] "gameActivity_broadcastUrl":[redacted]
3361       8637.99902344        [8928] "gameActivity_gamePresence":"",
3362       8637.99902344        [8928] "gameActivity_gameTitle":[redacted]
3363       8637.99902344        [8928] "gameActivity_multiplayerId":[redacted]
3364       8637.99902344        [8928] "gameActivity_productId":"Origin OFR 50 0002694"
```

Figure 1 - DebugView Application

We searched for all references to OutputDebugString function in the Origin installation directory.



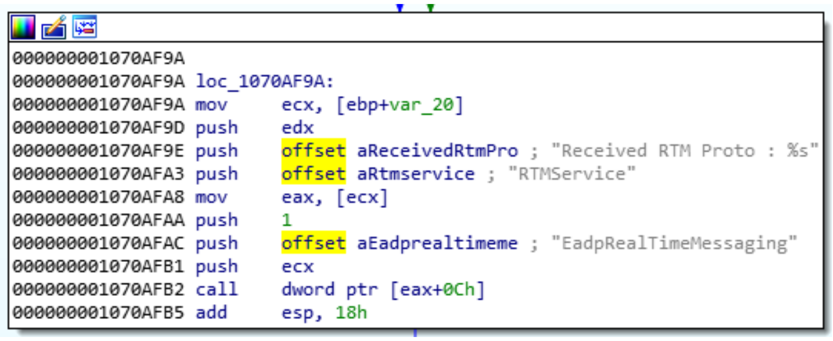
```
C:\Program Files (x86)\Origin>findstr /s /m OutputDebugString *.dll *.exe
d3dcompiler_47.dll
EALink.exe
EAProxyInstaller.exe
IGO32.dll
igo64.dll
legacyPM\CmdPortalClient.dll
legacyPM\EACore.dll
legacyPM\EAProxyInstaller.exe
legacyPM\QtCore4.dll
libEGL_Swift.dll
libGLESv2_Swift.dll
libmfxsw32.dll
Origin.exe
OriginClient.dll
OriginClientService.exe
OriginCrashReporter.exe
OriginER.exe
OriginThinSetupInternal.exe
OriginUninstall.exe
OriginWebHelperService.exe
Qt5Core.dll
Qt5WebEngineCore.dll
steam_api.dll
texttospeech\qtexttospeech_sapi.dll
twitchsdk_32_release.dll
x64\libmfxsw64.dll
x64\twitchsdk_64_release.dll

C:\Program Files (x86)\Origin>
```

Figure 2 - List of files with OutputDebugString references

After analyzing all these files, we focused on OriginClient.dll because this file contains most of the strings that were disclosing the sensitive information. The functions that send the information to the OutputDebugString function can be found at the following offsets:

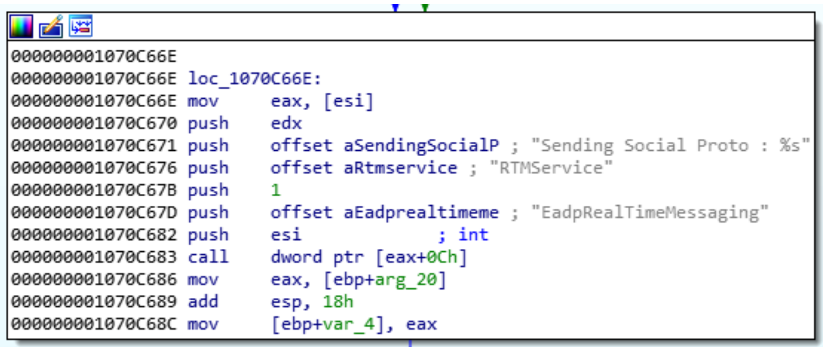
- 0x1070C2C0 - OriginClient.dll version 10.5.67.39484
- 0x1070AE80 - OriginClient.dll version 10.5.67.39484

The two images below show the blocks where the call path ends calling OutputDebugString.



```
000000001070AF9A
000000001070AF9A loc_1070AF9A:
000000001070AF9A mov     ecx, [ebp+var_20]
000000001070AF9D push    edx
000000001070AF9E push    offset aReceivedRtmPro ; "Received RTM Proto : %s"
000000001070AFA3 push    offset aRtmservice ; "RTMService"
000000001070AFA8 mov     eax, [ecx]
000000001070AFAA push    1
000000001070AFAC push    offset aEadprealtimeme ; "EadpRealTimeMessaging"
000000001070AFB1 push    ecx
000000001070AFB2 call    dword ptr [eax+0Ch]
000000001070AFB5 add     esp, 18h
```

Figure 3 - Block on the function that process messages have been received

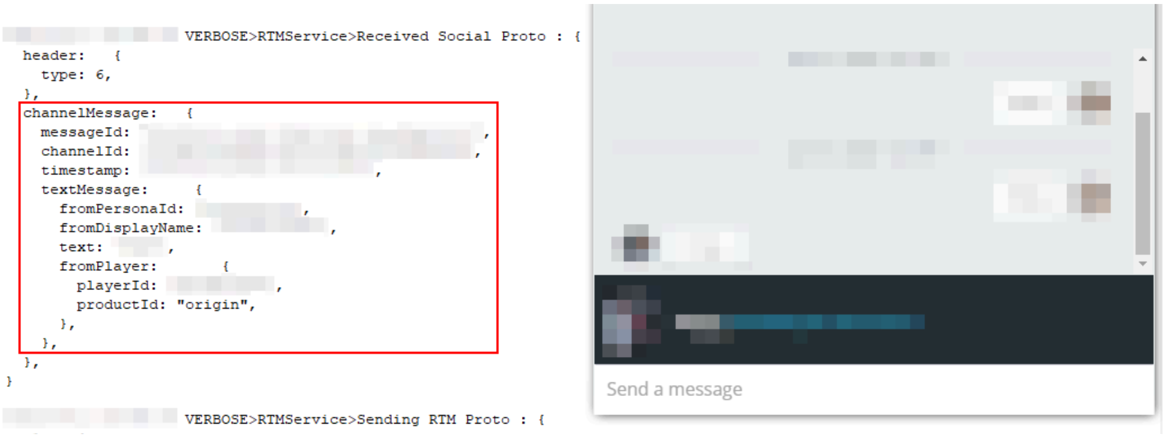


```
000000001070C66E
000000001070C66E loc_1070C66E:
000000001070C66E mov     eax, [esi]
000000001070C670 push    edx
000000001070C671 push    offset aSendingSocialP ; "Sending Social Proto : %s"
000000001070C676 push    offset aRtmservice ; "RTMService"
000000001070C67B push    1
000000001070C67D push    offset aEadprealtimeme ; "EadpRealTimeMessaging"
000000001070C682 push    esi             ; int
000000001070C683 call    dword ptr [eax+0Ch]
000000001070C686 mov     eax, [ebp+arg_20]
000000001070C689 add     esp, 18h
000000001070C68C mov     [ebp+var_4], eax
```

Figure 4 - Block on the function that process messages have been sent

Processing these messages could lead an attacker to obtain information such as the following:

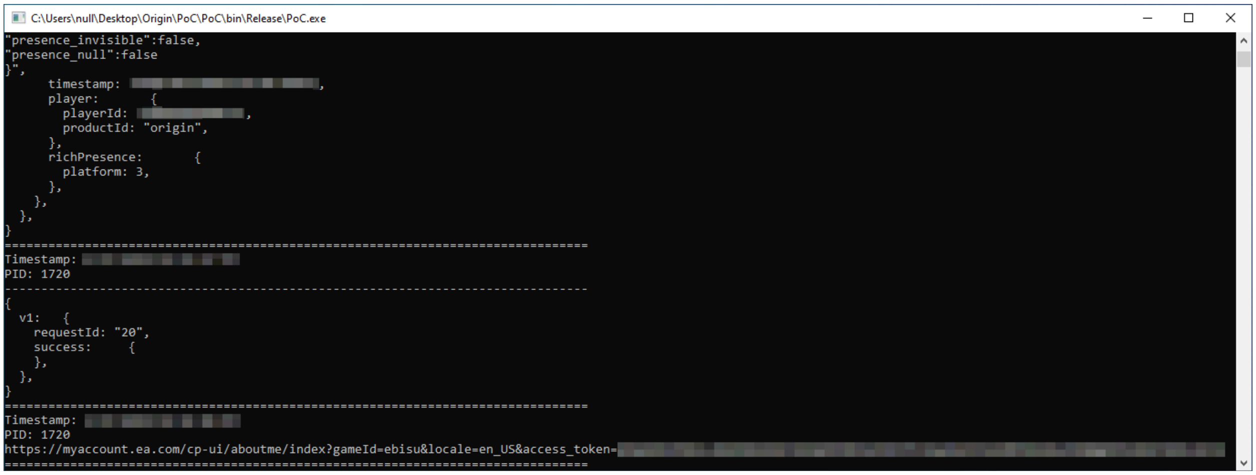
- username player ID
- username persona ID
- username display name
- username broadcast url
- username authentication token
- friend’s display name
- friend’s persona ID
- chat messages



```
[redacted] VERBOSE>RTMService>Received Social Proto : {
  header:   {
    type: 6,
  },
  channelMessage:   {
    messageId: [redacted],
    channelId: [redacted],
    timestamp: [redacted],
    textMessage:     {
      fromPersonaId: [redacted],
      fromDisplayName: [redacted],
      text: [redacted],
      fromPlayer:       {
        playerId: [redacted],
        productId: "origin",
      },
    },
  },
}
[redacted] VERBOSE>RTMService>Sending RTM Proto : {
```

Figure 5 - Chat messages display in the DebugView application

As it was mentioned before it’s simple for any application to process these messages. We developed a proof of concept application that is able to process Origin debug messages and even generate the URL containing the user token in order to log into the EA website without requiring any type of authentication.



```
"presence_invisible":false,
"presence_null":false
}",
      timestamp: [redacted],
      player:       {
        playerId: [redacted],
        productId: "origin",
      },
      richPresence:       {
        platform: 3,
      },
    },
  },
}
================================================================================
Timestamp: [redacted]
PID: 1720
--------------------------------------------------------------------------------
{
  v1:   {
    requestId: "20",
    success:     {
    },
  },
}
================================================================================
Timestamp: [redacted]
PID: 1720
https://myaccount.ea.com/cp-ui/aboutme/index?gameId=ebisu&locale=en_US&access_token=[redacted]
================================================================================
```

Figure 6 - Output from the proof of concept application

Once the attacker logs into the ea.com web site using the URL with the token it’s possible to access the following sites:

- myaccount.ea.com
- help.ea.com
- answers.ea.com
- origin.com

EA Security team confirmed that although access to this token will grant an attacker access to the user account, many of the sensitive operations are protected via two-factor authentication and other validation mechanisms.

Report Timeline

2020-04-22: Initial contact with the vendor via secure@ea.com.
2020-04-23: EA Security Team confirmed the reception of the email and requested the draft version of the advisory.
2020-04-24: A draft report with technical details and a proof of concept application was sent to the vendor.
2020-05-01: Immunity Inc. requests a status update.
2020-05-05: EA Security Team assigns the tracking ID PSECR-241 and informs their team will be investigating our report.
2020-05-11: Immunity Inc. requests a status update.
2020-05-11: EA Security Team informs they were able to reproduce the issue and are working on a fix.
2020-05-19: Immunity Inc. sent a request to Mitre for the CVE ID.
2020-05-19: Mitre assigns CVE-2020-13172.
2020-05-19: Immunity Inc. requests an estimated release date for the fix.
2020-05-19: EA Security Team informs that the product team is estimating the fix to be in production by end of June.
2020-06-08: Immunity Inc. requests the fix release date and sends an updated version of the advisory.
2020-06-15: Immunity Inc. requests a status update on the fix release date.
2020-06-17: EA Security Team confirmed the reception of the email and informs their team is working on the fix.
2020-06-22: Immunity Inc. notice that EA released Origin version 10.5.73.41506 and confirmed that the vulnerabilities was addressed.
2020-06-23: Immunity Inc. confirmed EA that due to the fact that the fix was already public the advisory report release date was set to June 24th.
2020-06-23: EA notify that the vulnerability has been addressed on Origin version 10.5.73.41506.
2020-06-24: Immunity Inc. proposed to reschedule the advisory release due to a potential issue with another advisory that is being process by EA Security Team.
2020-06-25: EA Security Team agreed to reschedule the advisory release.
2020-07-29: Immunity Inc. publish the advisory.

Disclaimer