VERSPRITE

RAZER SYNAPSE 3 VULNERABILITY ANALYSIS

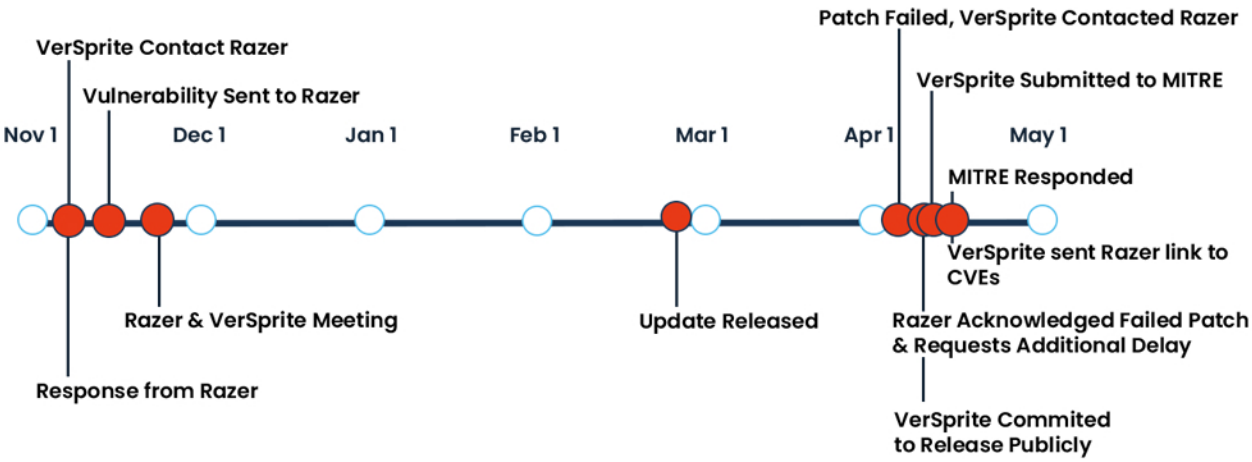
SECURITY RESEARCH

EXECUTIVE SUMMARY

Razers Synapse 3 product contains security related vulnerabilities that provided less privileged users the ability to write a file to any folder on disk. The vulnerability is within the improper usage of the Windows Registry, where improper permission assignment leads to local users having full control over multiple important Registry Keys relating to the Synapse 3 software suite. Local system services deployed via the Synapse 3 software suite, utilize the Registry Keys to build file name paths to store runtime logging information.

The initial impact of these vulnerabilities is a Denial of Service via system instability; however, full exploitation is not out of the realm of possibilities. These vulnerabilities should be considered a Medium risk level once patched and a High risk level until then. For more information about the disclosure process, see the Vulnerability Disclosure Timeline on the next page.

VULNERABILITY DISCLOSURE TIMELINE



Vendor Disclosure Timeline

11-06-2020	Contacted Razer and asked to be put in touch with a security resource for the disclosure process.
11-06-2020	Initial response from Razer was received.
11-11-2020	VerSprite provided report and vulnerability details via a report to Razer support.
11-17-2020	Razer & VerSprite had a disclosure meeting going over remediation steps.
02-25-2021	Razer released update to Synapse and remediated the issue.
04-07-2021	VerSprite performed Patch Verification and determined that a component was still vulnerable.
04-07-2021	VerSprite reached out to Razer to alert them that some components were still vulnerable.
04-08-2021	Razer responded acknowledging that their patch was incomplete and requested delay in notification to MITRE for CVE ID, until they released the patch Publicly at the end of April 2021.
04-08-2021	VerSprite responded with commitment to release schedule already presented and explained they will not delay public disclosure due to failed patch.
04-09-2021	VerSprite submitted initial vulnerability details to MITRE to acquire CVE-ID.
04-12-2021	MITRE responded with two CVE ID's(CVE-2021-30494 & CVE-2021-30493) for each vulnerability.
04-13-2021	VerSprite sent Razer link to publication of vulnerabilities

DIVING INTO THE RAZER SYNAPSE 3 VULNERABILITY

Razer, is a software and hardware-based company that offers products that are targeted towards consumers within the Video Game industry. Razer is responsible for creating the ***Razer Synapse 3*** software product. This product provides gamers with advanced capabilities surrounding modifications to hardware (i.e. computer mouse) behavior from a software level. VerSprite VS-Labs Security Researchers discovered security related issue within the ***Razer Synapse 3*** product itself. These issues are all among similar class of vulnerability, so, only one case is discussed, and the methodology can be applied to all the other similar cases.

Remediation for these vulnerabilities was performed on February 25th, 2021, when Razer released updates to the Synapse 3 Software suite where the vulnerabilities were mitigated. However, after internal verification of the patch provided by Razer, VerSprite Researchers have concluded that the patch was only a partial solution.

The ***RzSDKService.exe*** service binary, still interacted with a critical resource that had improper permissions assigned. Razer has acknowledged the failed patch and stated that they will work on a patch before the end of April 2021. For more information on the entire timeline, please refer to the Vendor Disclosure Timeline section.

What Is the Underlying Vulnerability?

The issues that plagued the **Razer Synapse 3** application are all centered around improper permissions assigned to critical resources, which relates to MITRE CWE: 732. The exact critical resources are Registry Key's located within the ***HKEY_LOCAL_MACHINE*** (HKLM), Hive, and the permissions assigned allow ***default users*** full control over the Registry

Keys. The exact Registry Keys are:

- **HKLM\SOFTWARE\WOW6432Node\Razer\ChromaBroadcast**
- **HKLM\SOFTWARE\WOW6432Node\Razer Chroma SDK**

The system services ***Razer Synapse Service.exe***, ***Razer Chroma SDK Server – RzSDKServer.exe***, and ***Razer Chroma SDK Service – RzSDKService.exe*** are all installed alongside the ***Razer Synapse 3*** software. These system services all run in a privileged context; this means that any operation performed is considered by default to be privileged. These system service perform privileged file write operations by obtaining a folder path from the **InstallPath** Registry Key entry from within the two respective Registry Key's ***\Razer\ChromaBroadcast*** and ***\Razer Chroma SDK***.

After obtaining the ***folder path*** value from the **InstallPath** entry, the system service then performs a concatenation by merging the follow strings together:

- **"folder path" + \Logs\ + ImageModuleName.exe + .log**

This string is broken down into four parts such as:

- Part One – Is the ***InstallPath*** entry Value.
  - ***folder path***
- Part Two – Hardcoded string that is appended to Part One.
  - ***\Logs***
- Part Three – This is the system service image module name.
  - **RzSDKService.exe**
- Part Four – Last appended string which signals the file is a ***Log*** file of some kind.
  - ***.log***

Each system service will have their specific image module name replaced within ***Part Three*** of the entire newly concatenated string.

With the knowledge that any ***default user*** can modify these ***InstallPath*** entries within each Registry Key, a malicious attacker could cause the system services to open a handle to a new file at any location on disk and cause an arbitrary write of semi-uncontrolled data. This mean that an unprivileged user could write any file anywhere on disk and the file name would be ***ImageModuleName.exe+.log***.

Understanding Impact and Limitations

Simply having the ability to create a file anywhere on disk seems quite powerful at first; however, this ability by itself is quite limited in this circumstance. The limitations are enforced because an attacker does not control the actual name of the newly created file or the extension. The second limitation is the data that is written to the newly created file, at first glance, does appear to be non-attacker controlled. With these issues in place abuse and impact comes down to a Denial of Service (DoS) if an attacker can somehow gain control of the actual file name itself.

Taking a Deeper Look at RzSDKService.exe

The ***Razer Chroma SDK Service – RzSDKService.exe***, obtains a handle to the Registry Key ***HKLM\SOFTWARE\WOW6432Node\Razer\ChromaBroadcast*** and enumerates the value associated with the entry, ***Install Path***.

The default entry value is: ***C:\Program Files (x86)\Razer\ChromaBroadcast***; however, default users have complete control over this Registry Key. This level of

control allows for malicious attackers to manipulate the ***InstallPath*** entry and replace the default value with a malicious value. Then during either a system reboot or a general restart of the system service ***RzSDKService.exe***, the system service will subsequently attempt to concatenate the following strings together and use the resulting file as a general runtime logging file.

Log File Concatenation
"C:\Program Files (x86)\Razer\ChromaBroadcast\" + "\Logs\" + "RzSDKService.exe" + ".log" "C:\Program Files (x86)\Razer\ChromaBroadcast\Logs\RzSDKService.exe.log"

Initial Discovery Methodology and Testing Observations

The initial discovery of this vulnerability is credited towards analyzing output captured with the tool ***Process Monitor*** (Procmon.exe) from ***SysInternals Suite***. Utilizing ***Procmon.exe***, we were able to identify the ***RegQueryValue*** operation and the subsequent ***CreateFile*** operation where the default value ***C:\Program Files(x86)\Razer\ChromaBroadcast*** within the ***InstallPath*** entry inside of the ***ChromaBroadcast*** subkey is extracted and used during the ***CreateFile*** operation.

By modifying the ***InstallPath*** entry we were able to capture the ***RzSDKService.exe*** system service attempting to perform a ***CreateFile*** operation; however, the operation returns with a simple ***PATH NOT FOUND*** as seen in the table on the next page.

Procmon Observation

Process Name: RzSDKService.exe
PID: 4268
Operation: RegQueryValue
Path: HKLM\SOFTWARE\WOW6432Node\Razer\ChromaBroadcast\InstallPath
Result: SUCCESS
Detail: Type: REG_SZ, Length: 58, Data: TestMe-ChromaBroadcast-Case1

Process Name: RzSDKService.exe
PID: 4268
Operation: CreateFile
Path: C:\WINDOWS\SysWOW64\TestMe-ChromaBroadcast-Case1\Logs\RzSDKService.exe.log
Result: PATH NOT FOUND
Detail: Desired Access: Generic Read/Write, Disposition: OpenIf, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, Share Mode: Read, Write, Allocation Size: 0

Achieving Controlled File Creation with Arbitrary Data

An attacker can modify the I***nstallPath*** entries value to ***C:\Users\User\Desktop\TestMe-ChromaBroadcast-Case1*** and then use the ***CreateSymlink.exe*** tool, to create a symlink that points to a path within a privileged folder, such as ***C:\Windows\System32***.

The exact steps are outlined in the tables below for reference.

Creation of Symbolic Link

C:\Users\User\Desktop\Release>CreateSymlink.exe C:\Users\User\Desktop\TestMe-ChromaBroadcast-Case1\Logs\RzSDKService.exe.log C:\Windows\System32\TestMe-RzSDKService-Case1.txt
Opened Link \RPC Control\RzSDKService.exe.log -> \??\C:\Windows\System32\TestMe-RzSDKService-Case1.txt: 00000158
Press ENTER to exit and delete the symlink

Restarting the service and monitoring with Procmon

Process Name: RzSDKService.exe
PID: 9748
Operation: RegQueryValue
Path: HKLM\SOFTWARE\WOW6432Node\Razer\ChromaBroadcast\InstallPath
Result: SUCCESS
Detail: Type: REG_SZ, Length: 102, Data: C:\Users\User\Desktop\TestMe-ChromaBroadcast-Case1

Process Name: RzSDKService.exe
PID: 9748
Operation: CreateFile
Path: C:\Users\User\Desktop\TestMe-ChromaBroadcast-Case1\Logs\RzSDKService.exe.log
Result: REPARSE
Detail: Desired Access: Generic Read/Write, Disposition: OpenIf, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0, OpenResult: <unknown>

Process Name: RzSDKService.exe
PID: 9748
Operation: CreateFile
Path: C:\Windows\System32\TestMe-RzSDKService-Case1.txt
Result: SUCCESS
Detail: Desired Access: Generic Read/Write, Disposition: OpenIf, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: 0, OpenResult: Opened

Reviewing Contents of "TestMe-RzSDKService-Case1.exe" and its new permissions

```
C:\Users\User\Desktop\Release>type C:\Windows\System32\TestMe-RzSDKService-Case1.txt
[2020-11-10 16:08:29][Info][ChromaBroadcastManager][END] ********** C:\Program Files (x86)\Razer Chroma
SDK\bin\RzSDKService.exe(2864) **********
[2020-11-10 16:08:29][Info][ChromaBroadcastManager][START] ********** C:\Program Files (x86)\Razer Chroma
SDK\bin\RzSDKService.exe(9748) **********

C:\Users\User\Desktop\Release>C:\Windows\System32\icacls.exe C:\Windows\System32\TestMe-RzSDKService-Case1.txt
C:\Windows\System32\TestMe-RzSDKService-Case1.txt NT AUTHORITY\SYSTEM:(I)(F)
    BUILTIN\Administrators:(I)(F)
    BUILTIN\Users:(I)(RX)
    APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
    APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(I)(RX)

    Successfully processed 1 files; Failed processing 0 files
```

Vulnerability Remediation

VerSprite's Research and Development Team, VS-Labs, recommends restricting access to all critical local resources to only privileged users, if possible. This will prevent unauthorized or lower privileged users from modifying these critical resources.