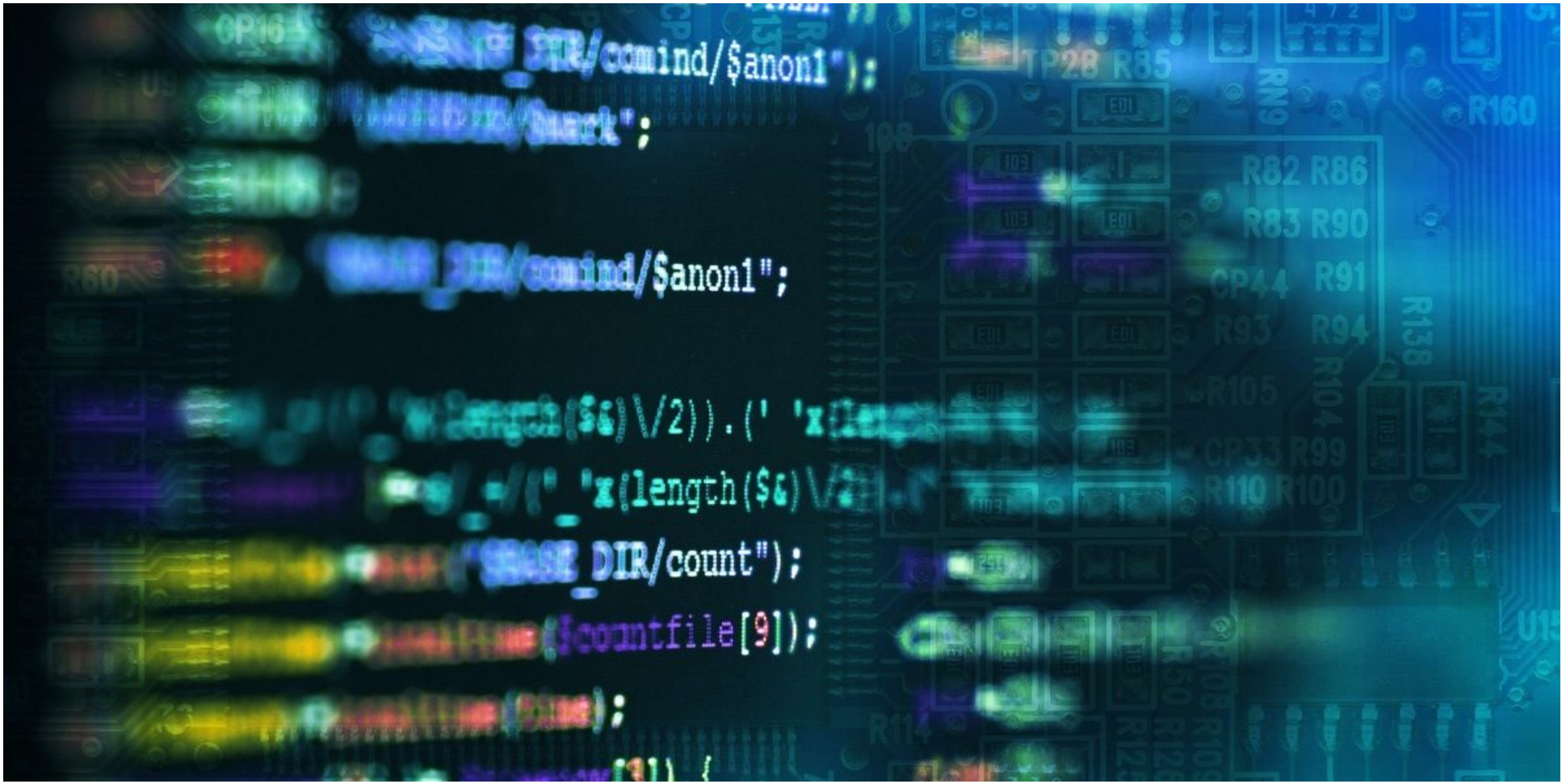
PuzzleMaker attacks with Chrome zero-day exploit chain

securelist.com/puzzlemaker-chrome-zero-day-exploit-chain/102771



On April 14-15, 2021, Kaspersky technologies detected a wave of highly targeted attacks against multiple companies. Closer analysis revealed that all these attacks exploited a chain of Google Chrome and Microsoft Windows zero-day exploits. While we were not able to retrieve the exploit used for remote code execution (RCE) in the Chrome web browser, we were able to find and analyze an elevation of

privilege (EoP) exploit that was used to escape the sandbox and obtain system privileges.

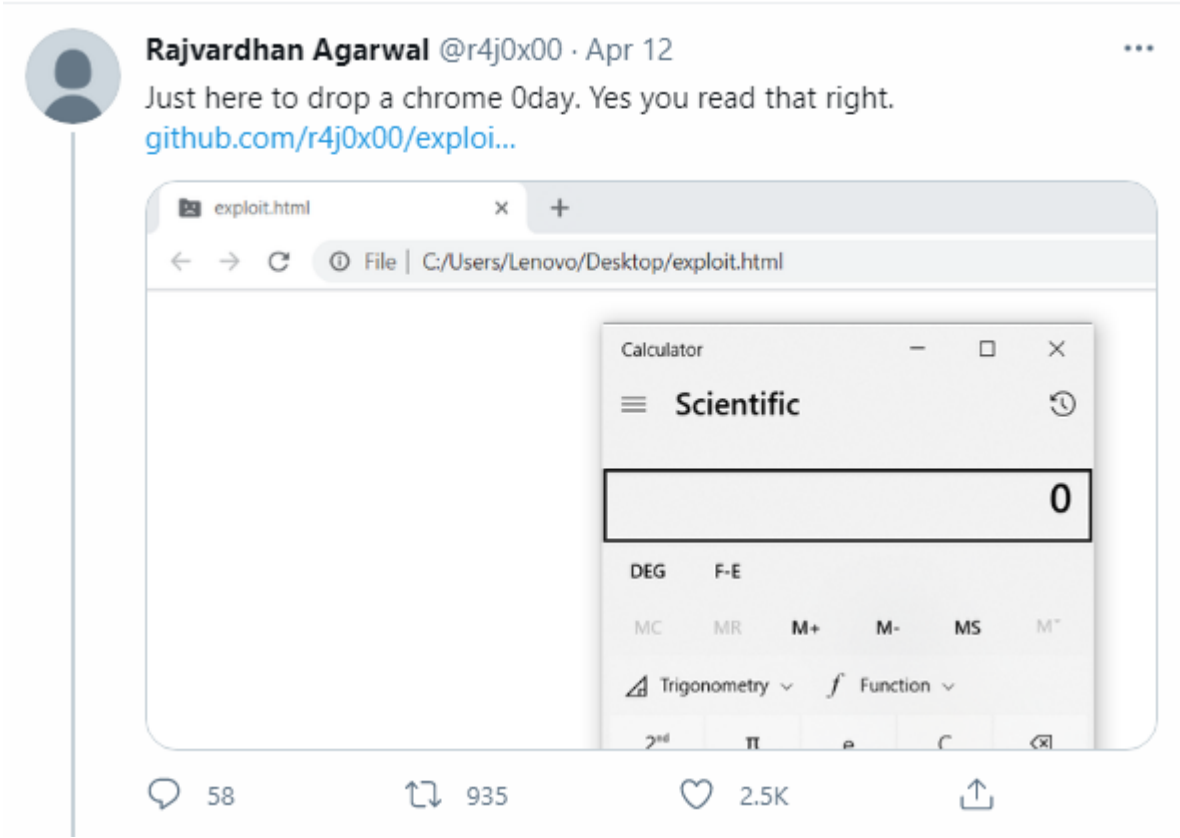
The elevation of privilege exploit was fine-tuned to work against the latest and most prominent builds of Windows 10 (17763 – RS5, 18362 – 19H1, 18363 – 19H2, 19041 – 20H1, 19042 – 20H2) and it exploits two distinct vulnerabilities in the Microsoft Windows OS kernel. On April 20, 2021, we reported these vulnerabilities to Microsoft and they assigned CVE-2021-31955 to the information disclosure vulnerability and CVE-2021-31956 to the elevation of privilege vulnerability. Both vulnerabilities were patched on June 8, 2021, as a part of the June Patch Tuesday.

Remote code execution exploit

All of the observed attacks were conducted through Chrome browser. Unfortunately, we were unable to retrieve the JavaScript with full exploit code, but the timeframe of attacks and events preceding it led us to suspect one particular vulnerability.

On April 6-8, 2021 the Pwn2Own competition took place. This is a computer hacking contest where the Google Chrome web browser was one of the targets. According to the ZDI (Zero Day Initiative, the organizer of Pwn2Own) website, one participating team was able to demonstrate a successful exploitation of the Chrome renderer process using a Typer Mismatch bug.

On April 12, 2021, the developers of Chromium committed two (issue 1196683, issue 1195777) Typer-related bug fixes to the open-source repository of V8 – a JavaScript engine used by Chrome and Chromium web browsers. One of these bug fixes (issue 1196683) was intended to patch a vulnerability that was used during Pwn2Own, and both bug fixes were committed together with regression tests – JavaScript files to trigger these vulnerabilities. Later on the same day, a user with the Twitter handle @r4j0x00 published a working remote code execution exploit on GitHub, targeting an up-to-date version of Google Chrome. That exploit used a vulnerability from issue 1196683 to execute a shellcode in the context of the browser renderer process.



Screenshot of tweet with Chrome zero-day published on April 12, 2021

The published exploit didn't contain a sandbox escape exploit and was therefore intended to work only when the browser was launched with the command line option *–no-sandbox*.

On April 13, 2021, Google released Chrome update 89.0.4389.128 for Windows, Mac and Linux with a fix for two vulnerabilities; CVE-2021-21220 (used during Pwn2Own) was one of them.

Some of our customers who were attacked on April 14-15, 2021, already had their Chrome browser updated to 89.0.4389.128, and that's why we think the attackers didn't use CVE-2021-21220 in their attacks.

On April 14, 2021, Google released Chrome update 90.0.4430.72 for Windows, Mac and Linux with a fix for 37 vulnerabilities. On the same day, a new Chrome exploit was presented to the public.

avboy1337 / 1195777-chrome0day

Notifications | Star 331 | Fork 121

Code | Issues 1 | Pull requests 1 | Actions | Projects | Wiki | Security | ...

Create 1195777.html

Browse files

main

avboy1337 committed on Apr 14 Verified 1 parent c69f29d commit e2a004a0a6cc9bbd3780d6bd585eee1131adbb4f

Showing 1 changed file with 103 additions and 0 deletions. Unified Split

103 ■■■■■ 1195777.html

```
@@ -0,0 +1,103 @@
+ <script>
+     function gc() {
+         for (var i = 0; i < 0x80000; ++i) {
+             var a = new ArrayBuffer();
+         }
+     }
+     let shellcode = [0xFC, 0x48, 0x83, 0xE4, 0xF0, 0xE8, 0xC0, 0x00, 0x00, 0x00, 0x41, 0x51, 0x41, 0x50, 0x52, 0
+         0x56, 0x48, 0x31, 0xD2, 0x65, 0x48, 0x8B, 0x52, 0x60, 0x48, 0x8B, 0x52, 0x18, 0x48, 0x8B, 0x52,
+         0x20, 0x48, 0x8B, 0x72, 0x50, 0x48, 0x0F, 0xB7, 0x4A, 0x4A, 0x4D, 0x31, 0xC9, 0x48, 0x31, 0xC0,
+         0xAC, 0x3C, 0x61, 0x7C, 0x02, 0x2C, 0x20, 0x41, 0xC1, 0xC9, 0x0D, 0x41, 0x01, 0xC1, 0xE2, 0xED,
+         0x52, 0x41, 0x51, 0x48, 0x8B, 0x52, 0x20, 0x8B, 0x42, 0x3C, 0x48, 0x01, 0xD0, 0x8B, 0x80, 0x88,
```

```
12  +       0x00, 0x00, 0x00, 0x48, 0x85, 0xC0, 0x74, 0x67, 0x48, 0x01, 0xD0, 0x50, 0x8B, 0x48, 0x18, 0x44,
```

Screenshot of GitHub repository with Chrome zero-day published on April 14, 2021

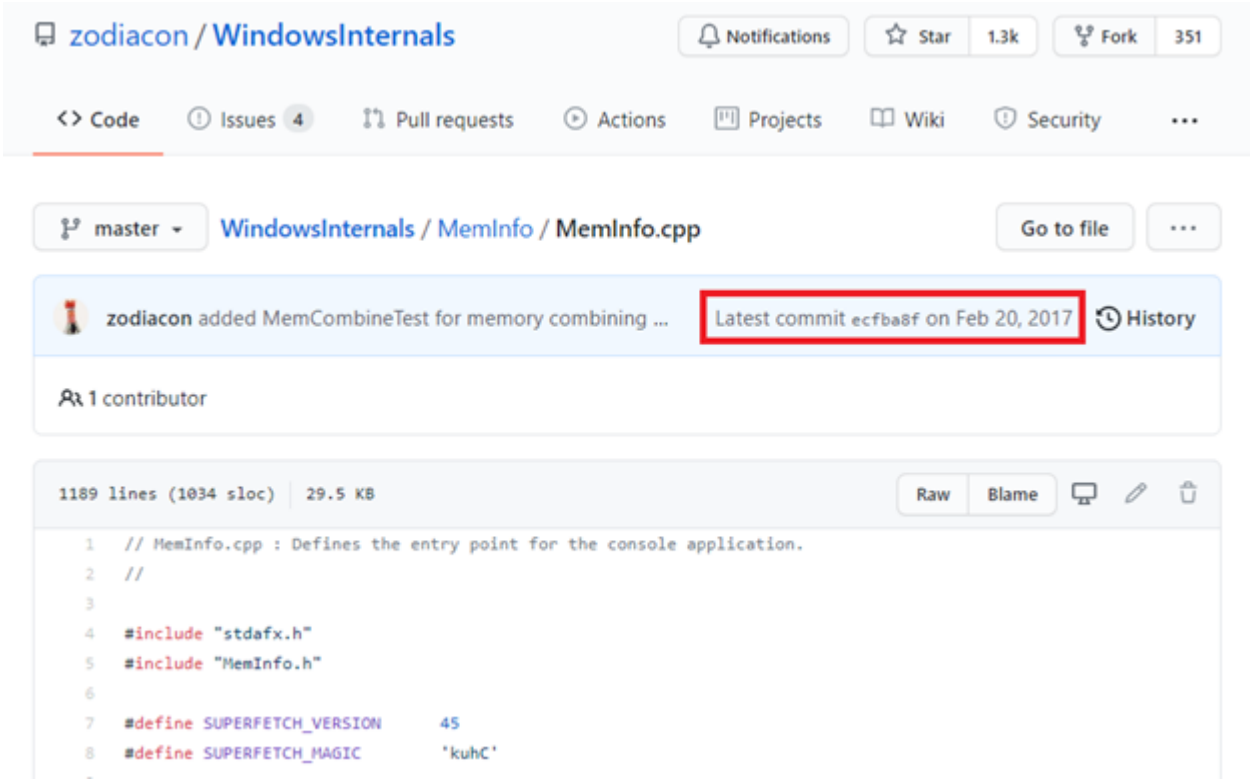
This newly published exploit used a vulnerability from issue 1195777, worked on the newly released Chrome 90.0.4430.72, and was fixed as CVE-2021-21224 only a few days later, on April 20, 2021.

We suspect the attackers were also able to use this JavaScript file with regression test to develop the exploit (or acquire it from someone else) and were probably using CVE-2021-21224 in their attacks.

Elevation of privilege exploit

CVE-2021-31955 is an information disclosure vulnerability in ntoskrnl.exe. The vulnerability is affiliated with a Windows OS feature called SuperFetch. It was introduced in Windows Vista and is aimed to reduce software loading times by pre-loading commonly used applications into memory. For SuperFetch purposes the function *NtQuerySystemInformation* implements a special system information class *SystemSuperfetchInformation*. This system information class incorporates more than a dozen of different SuperFetch information classes. The vulnerability lies in the fact that data returned by the *NtQuerySystemInformation* function for the SuperFetch information class *SuperfetchPrivSourceQuery* contains EPROCESS kernel addresses for currently executed processes.

It’s noteworthy that this vulnerability can be observed in code that was available on GitHub for a few years before we caught it in the wild and Microsoft patched it.



CVE-2021-31955 can be observed in the source code of the MemInfo utility

The other vulnerability, CVE-2021-31956, is a heap-based buffer overflow in ntfs.sys. The function *NtfsQueryEaUserEaList* processes a list of extended attributes for the file and stores the retrieved values to buffer. This function is accessible via *ntoskrnl* syscall and among other things it's possible to control the size of the output buffer. If the size of the extended attribute is not aligned, the function will calculate a padding and the next extended attribute will be stored 32-bit aligned. The code checks if the output buffer is long enough to fit the extended attribute with padding, but it doesn't check for possible integer-underflow. As a result, a heap-based buffer overflow can happen.

```
1    for ( cur_ea_list_entry = ea_list; ; cur_ea_list_entry = next_ea_list_entry )
2    {
```

```
3      ...
4      out_buf_pos = (DWORD *)(out_buf + padding + occupied_length);
5      if ( NtfsLocateEaByName(eas_blocks_for_file, eas_blocks_size, &name, &ea_block_pos) )
6      {
7    ea_block = eas_blocks_for_file + ea_block_pos;
8    ea_block_size = ea_block->DataLength + ea_block->NameLength + 9;
9    if ( ea_block_size <= out_buf_length - padding ) // integer-underflow is possible
10   {
11     memmove(out_buf_pos, (const void *)ea_block, ea_block_size); // heap buffer overflow
12     *out_buf_pos = 0;
13   }
14    }
15    else
16    {
17   ...
18    }
19    ...
20    occupied_length += ea_block_size + padding;
21    out_buf_length -= ea_block_size + padding;
```

```
22    padding = ((ea_block_size + 3) & 0xFFFFFFFC) - ea_block_size;
23    ...
24  }
25
26
27
28
29
```

Pseudo-code for vulnerable code in function NtfsQueryEaUserEaList

The exploit uses CVE-2021-31956 along with Windows Notification Facility (WNF) to create arbitrary memory read and write primitives. We are planning to publish more information about this technique in the future.

As the exploit uses CVE-2021-31955 to get the kernel address of the EPROCESS structure, it is able to use the common post exploitation technique to steal SYSTEM token. However, the exploit uses a rarely used “PreviousMode” technique instead. We have seen this technique used by the CHAINSHOT framework and even made a presentation about it at CanSecWest/BlueHat in 2019. The exploit uses this technique to inject a malware module into the system process and execute it.

Malware modules

Besides the aforementioned exploits, the full attack chain consists of four additional malware modules, which will be referred to as:

- Stager
- Dropper
- Service

- Remote shell

The stager module is used to notify that exploitation was successful. It also downloads and executes a more complex malware dropper module from a remote server. Each stager module is delivered to the victim with a personalized configuration blob that defines the C&C URL, Session ID, keys to decrypt the next stage of malware, and other information.

All the stager module samples that we've discovered so far were configured to use the same URL address – hxxps://p{removed}/metrika_upload/index.php – to download the encrypted malware dropper module.

We believe there is a chance that the remote code execution JavaScript exploit was also hosted on the same legitimate-looking geopolitical news portal, but we found no evidence of a classic watering hole attack. The victimology suggests a highly targeted delivery of exploits.

The dropper module is used to install two executables that pretend to be legitimate files belonging to Microsoft Windows OS. One of these files (%SYSTEM%\WmiPrvMon.exe) is registered as a service and is used as a launcher for the second executable. This second executable (%SYSTEM%\wmimon.dll) has the functionality of a remote shell and can be considered the main payload of the attack. We couldn't find any similarities between this and other known malware.

The remote shell module has a hardcoded URL of the C&C server inside (media-seoengine[.]com). All the communication between C&C server and client is authorized and encrypted. The remote shell module is able to download and upload files, create processes, sleep for specified amounts of time and delete itself from the compromised machine.

None of the artifacts we analyzed appear to have strong connections to any known threat actors. The only similarity to CHAINSHOT we observed is the "PreviousMode" technique, although this is publicly known and may be used by various groups. We are calling the threat actor behind these attacks PuzzleMaker.

Kaspersky products detect this exploit and malware modules with the verdicts:

- PDM:Exploit.Win32.Generic
- PDM:Trojan.Win32.Generic
- UDS:DangerousObject.Multi.Generic

Kaspersky products detected these attacks with the help of the Behavioral Detection Engine and the Exploit Prevention component. Over the past few years, we have built a multitude of exploit protection technologies into our products that have detected many zero-days, repeatedly proving their effectiveness. We will continue to improve defenses for our users by enhancing technologies and working with third-party vendors to patch vulnerabilities, making the internet more secure for everyone.

More information about these attacks and the actor behind them is available to customers of the Kaspersky Intelligence Reporting service. Contact: intelreports@kaspersky.com.

Kaspersky would like to thank Microsoft for their prompt analysis of the report and patches.

IoCs

media-seoengine[.]com

%SYSTEM%\WmiPrvMon.exe

MD5 09A5055DB44FC1C9E3ADD608EFFF038C
SHA-1 BFFA4462901B74DBFBFFAA3A3DB27DAA61211412
SHA-256 982F7C4700C75B81833D5D59AD29147C392B20C760FE36B200B541A0F841C8A9

%SYSTEM%\wmimon.dll

MD5 D6B850C950379D5EE0F254F7164833E8
SHA-1 E63ED3B56A5F9A1EA5C92D3D2444196EA13BE94B
SHA-256 8A17279BA26C8FBE6966EA3300FDEFB1ADAE1B3ED68F76A7FC81413BD8C1A5F6

PuzzleMaker attacks with Chrome zero-day exploit chain