Team Name: Noble Team

Current Leader: Tanya Malik

Members: Melanie Brown, Marwan Elashry, Viren Kumar, Tanya Malik, Emily Nolan, and Lucas Scharf

**Melanie Brown:** I was able to get the man-in-the-middle attack to work. I was able to connect to the internet and Cortana to the internet while the MITM attack was going on. I continued to familiarize myself with cortana looking at previous findings and am currently looking into certificates for BurpSuite.

**Marwan Elashry:** I got my VMs and tools to work. I also researched relevant CVEs and attempted to recreate some of last year’s findings with some success, but got issues with some of them too.

**Viren Kumar:** I was able to get man in the middle attack working multiple times, no issue. I was also able to make a list of things that the last year group completed and attempted to recreate those findings with some success. I also wrote about a brower CVE.

**Tanya Malik:**

I compiled a list of some of the findings from last year and determined the list of results needed to replicate for the project.

**Emily Nolan:**

I verified and compiled a list of some of the findings from last year. I researched some older browser CVEs and read through last year's list of relevant CVEs and wrote 2 of the ones in this document. I also now have all of the tools working properly in the VMs.

**Lucas Scharf:**

I caught everyone up on the tools and steps required to do a MITM attack. I verified results from last year in Cortana’s local files. I researched CVE, installed Fiddler, and am currently working on capturing Cortana packets and voice packets to verify more results.

**Requested Written from last Meeting**

CVE 1: CVE-2021-29944

This CVE is rated a 6.1, medium, on the 3.0 version and had a partial impact on integrity and a medium impact on the access complexity. This vulnerability was found to only be present in Firefox Android. It was susceptible to cross site scripting and allowed for HTML injection when the website was in a reader view. The CVE description suggests that an HTML injection is unlikely because of other security policies in place but there is still a possibility. This was fixed though the vendor over 1 year ago.

CVE 2: CVE-2021-30614

There can be software that writes data past the end of the buffer, or before the beginning of the buffer. This typically occurs when the pointer or its index is incremented or decremented to a position beyond the bounds of the buffer or when pointer arithmetic results in a position outside of the valid memory location. This may result in corruption of sensitive information, a crash, or code execution among other things.

CVE 3: CVE-2022-30127

This CVE is rated an 8.8, high, on the 3.0 version and has an elevation of privilege vulnerability. Its CWE explains that there is the possibility for a race condition because there is a code sequence in the program that runs at the same time as another sequence of code. While this might not normally be a high issue, both of these sequences require exclusive access to a shared resource temporarily. This was fixed earlier this year through a vendor patch.

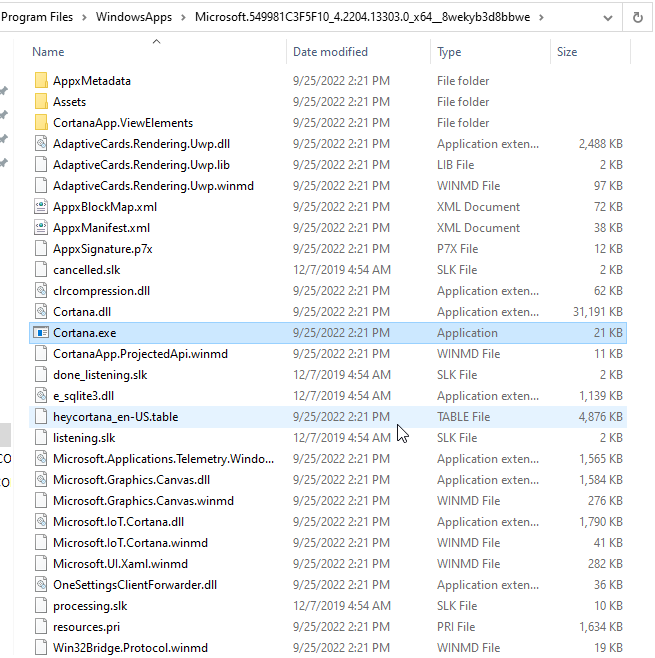
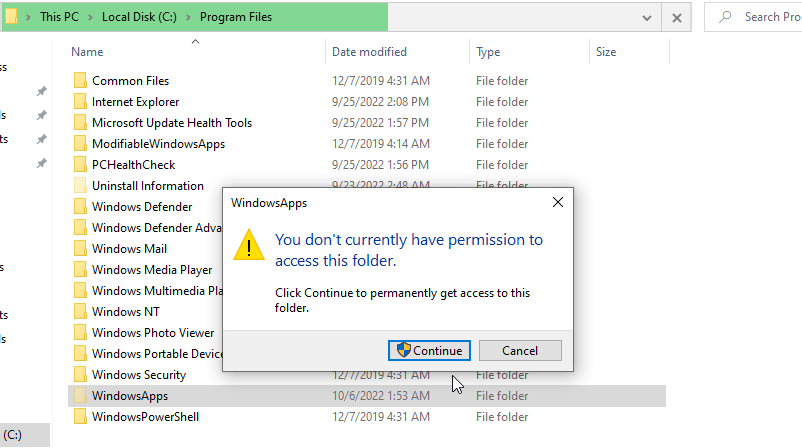
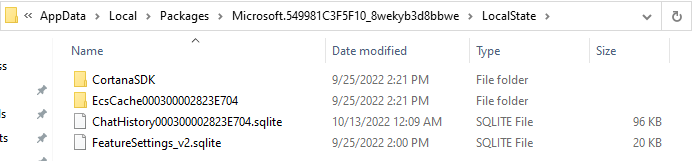
CVE 4: CVE-2019-5789

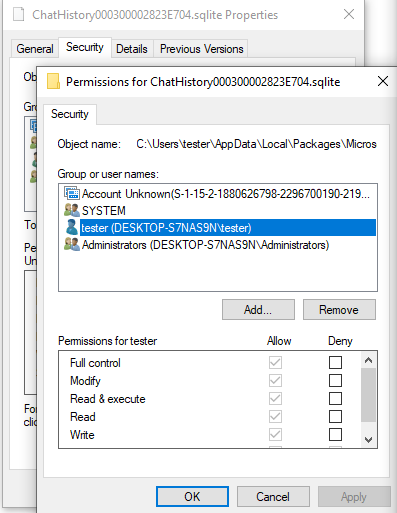
The CVE against google chrome browsers allow the attacker to gain complete access over confidentiality, integrity, and availability. This attack was an integer overflow that lead to use-after-free in WebMIDI on windows machines. Use-After-Free (UAF) is a vulnerability related to incorrect use of dynamic memory during program operation. This basically means that an attacker can remotely cause data corruption, program crashes, and arbitrary code execution.

<https://www.cvedetails.com/cve/CVE-2019-5789/>

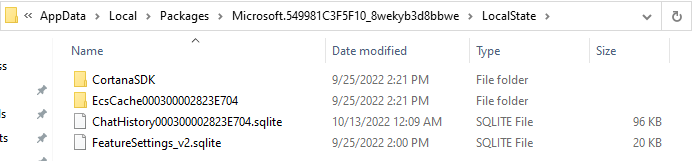
<https://encyclopedia.kaspersky.com/glossary/use-after-free/>

**Findings from Last Year:**

1. **File Structure:** 
   1. It was discovered that the primary directory of the Cortana program could be found in a subdirectory under Program Files, which contained the Cortana application, various dynamically-linked libraries (DLLs), and other files which are required for functionality
      1. 
   2. A subdirectory found in a user’s AppData folder appeared to contain Cortana’s local data for an individual user.
      1. 
      2. ChatHistory file contains all of the conversation history & does not require administrative privileges to access



* + 1. This directory was largely unremarkable, aside from one ChatHistory file located in the LocalState directory, which contained a record of all Cortana exchanges stored as an SQLite database.



* + 1. It was able to be read with any simple program that supports SQLite files.
    2. The database was found to contain the following information for each request and response: message type (request or response); timestamp; globally unique identifier (GUID); JSON command data (for responses only); plain text of request or response; and additional identifier information (for responses only).

1. **Code Analysis**
   1. Disassembling in Ghidra revealed that this file contained an incredible 31,322 functions, suggesting that this module was responsible for the majority of Cortana’s functionality.
      1. Unable to get working
2. **Cortana Network Communications**
   1. Fiddler was used to capture and analyze network traffic, while Burp Suite was used to test the effects of modifications to that traffic separately from Fiddler.
   2. Cortana was determined to communicate to a specific Microsoft domain, nam.api.cortana.ai, in nearly every instance other than for remote diagnostics;
   3. in those cases, settings-win.data.microsoft.com is contacted instead.
   4. In all instances, communications are encrypted using the Transport Layer Security (TLS) protocol
   5. JSON data is represented as attribute:value pairs as well as arrays or lists of pairs, meaning it is often in a human-readable format. If the query was spoken by the user instead of typed into Cortana’s “chat” interface, then the audio data is transmitted as SILK version 3 audio data.
3. **Authenticity and Integrity of Network Communications**
   1. During our experimentation with Cortana, it was discovered that the JSON data in the request and response packets can be freely modified and still be processed by Cortana.
   2. For this to work, the Windows virtual machine used for testing had to be configured to use the IP address of another virtual machine running Burp Suite as a proxy server in the system settings.
   3. SILK version 3 audio data, while unable to be decoded and played by Burp Suite, can be recovered more painstakingly using a tool such as Wireshark that can capture and save individual network traffic packets.