**Title:** Information Security Exploit in Action

**Assignment:** Assignment 1 Deliverable 3: Report

**Team Number:** 4

**Team:** Alexander Ponomaroff, Mamadou Diallo, Nicholas Krilis

**Course Code:** BTN710 NBB

Table of Contents

[Introduction 2](#_Toc24412130)

[Section 1: The Exploit - Summary 3](#_Toc24412131)

[Brief Description 3](#_Toc24412132)

[Operating System (OS) 3](#_Toc24412133)

[Protocols/Services/Applications 3](#_Toc24412134)

[Variants 4](#_Toc24412135)

[References 4](#_Toc24412136)

[Section 2: The Attack 5](#_Toc24412137)

[Description and Diagram of Network 5](#_Toc24412138)

[Protocol/Service Description 5](#_Toc24412139)

[How the Exploit Works and Description of the Attack 6](#_Toc24412140)

[Signature of the Attack 10](#_Toc24412141)

[How to Protect Against It 10](#_Toc24412142)

[Remediated System Test 11](#_Toc24412143)

[Section 3: Security Policy 11](#_Toc24412144)

[Conclusion 12](#_Toc24412145)

[References 13](#_Toc24412146)

# Introduction

Since the beginning of the Windows operating system, attackers have been working on different ways to exploit the system through escalation of privilege vulnerabilities. Escalation of privilege is an exploit that allows the attacker to gain access to administrative privileges on a system and gain access to information that is otherwise protected and can only be accessed by specific users who have permissions to access such information. Many different ways to perform such an exploit exist on different versions of the Windows OS. One of the most popular ways to perform such an exploit still exists today and only recently Microsoft has started to release updates that address this threat and try to mitigate it. In this report, we will be addressing this specific way to exploit Windows and gain escalation of privileges.

# Section 1: The Exploit - Summary

## Brief Description

Many versions of the Windows operating system contain a vulnerability that allows an attacker to escalate their privileges through the Windows Text Service Framework, and more specifically the mysterious CTF protocol that Windows Text Services Framework utilizes. With this vulnerability, a user with standard login privileges can escalate their privileges to have administrator access to the system.

This is especially problematic when the attacker is on an enterprise network in a company with many connected accounts. The attacker can use a tool called CTF Tool that is readily available on GitHub. This allows the individual to gain access to sensitive data stored on accounts on an enterprise network and gain administrative privileges on computers, where the attacker can use a standard login.

## Operating System (OS)

The Windows Text Services Framework exploit through the CTF protocol has been known to exist on almost all Windows operating systems. However, the recent concerns surrounding vulnerabilities in the TSF caused the CVE Details community to raise attention towards this vulnerability on September 11, 2019. It is stated that the affected versions of Windows include: Windows 10, 7, 8.1, Rt 8.1, and Windows Server 2008, 2012, 2016, 2019. Windows 10 has the most affected versions, including: 1607, 1703, 1709, 1803, 1809, and 1903. However, in this report, we will be exploring the exploit in Windows 7 Professional with the Service Pack 1 update, which is also affected.

## Protocols/Services/Applications

* Text Services Framework (TSF)
* CTF Protocol
* CTF Tool

## Variants

There are many different vulnerabilities within Windows and applications within Windows that get discovered quite often. Most, however, are very difficult to exploit and don’t have readily available tools online that allow for beginner level hackers to exploit easily. A lot of these variants of escalation of privilege exploits can be found through CVE Details website. For example, CVE-2019-0735 documented vulnerability states that there is an elevation of privilege problem when Windows Client Server Run-Time Subsystem fails to handle objects in memory. Another example can be seen in the CVE-2018-6674, which documents another escalation of privilege vulnerability that exists through a vulnerability in the McTray.exe within the McAfee VirusScan Enterprise 8.8. All of these exploits follow a similar process of exploiting vulnerabilities within different frameworks and protocols to gain higher than permitted privileges on a system or application. Although the process is similar, the main difference between these exploits lies in what vulnerability is being exploited. A vulnerability may exist within an application, within a Windows service, within a protocol used by a service. All of these variants target different applications, services, and protocols and vulnerabilities within them. However, the outcome is the same, gaining higher system privileges.

## References

Link to the vulnerability info: <https://www.cvedetails.com/cve/CVE-2019-1235/>

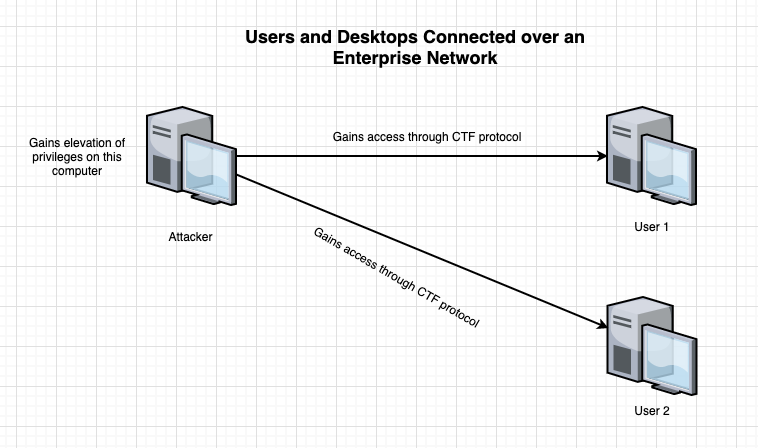
Link to the tool used to exploit this vulnerability: <https://github.com/taviso/ctftool>

Link to the blog that talks about exploit: <https://blog.stealthbits.com/using-ctftool-exe-to-escalate-privileges-by-leveraging-text-services-framework-and-mitigation-processes-and-steps/>

Link to the blog from the creator of CTF Tool: <https://googleprojectzero.blogspot.com/2019/08/down-rabbit-hole.html>

# Section 2: The Attack

## Description and Diagram of Network



The exploit uses flaws in Microsoft Windows Text Services Framework and the CTF protocol that it utilizes to launch child processes that allows the user to escalate privileges through the use of CTF Tool. The exploit essentially only needs one tool for the attacker to succeed and that is the CTF Tool. This tool has been developed solely for the use to exploit the weaknesses found in the TSF. This is not necessarily an exploit that is used over a network; however, it can be. If an attacker gains access to an account with standard permissions on a computer, he can escalate these permissions to system level permissions and gain full access to all accounts and all information that is stored on the computer. However, if an attacker gains access to a computer that is connected to an enterprise network of users and other computers, he can also exploit the CTF protocol vulnerabilities and gain access to all vulnerable users over the enterprise network.

In order for this exploit to work efficiently there are some requirements that must be met. The operating system must have Visual C++ Redistributable x86, a language pack that uses IME (Input Method Editor) and disable Windows Defender Realtime Protection. After ensuring these requirements are met the user is all set to go and exploit the process.

## Protocol/Service Description

Text Services Framework (TSF) is a system service for Windows operating system. It is responsible for natural processing and text input on Windows. TSF allows applications to receive text input from the users without having to deal with the hardware that gathered the input. It first appeared on Windows XP, and all future Windows operating systems used this framework, which allowed for a vulnerability in this framework to affect almost all Windows operating systems since Windows XP.

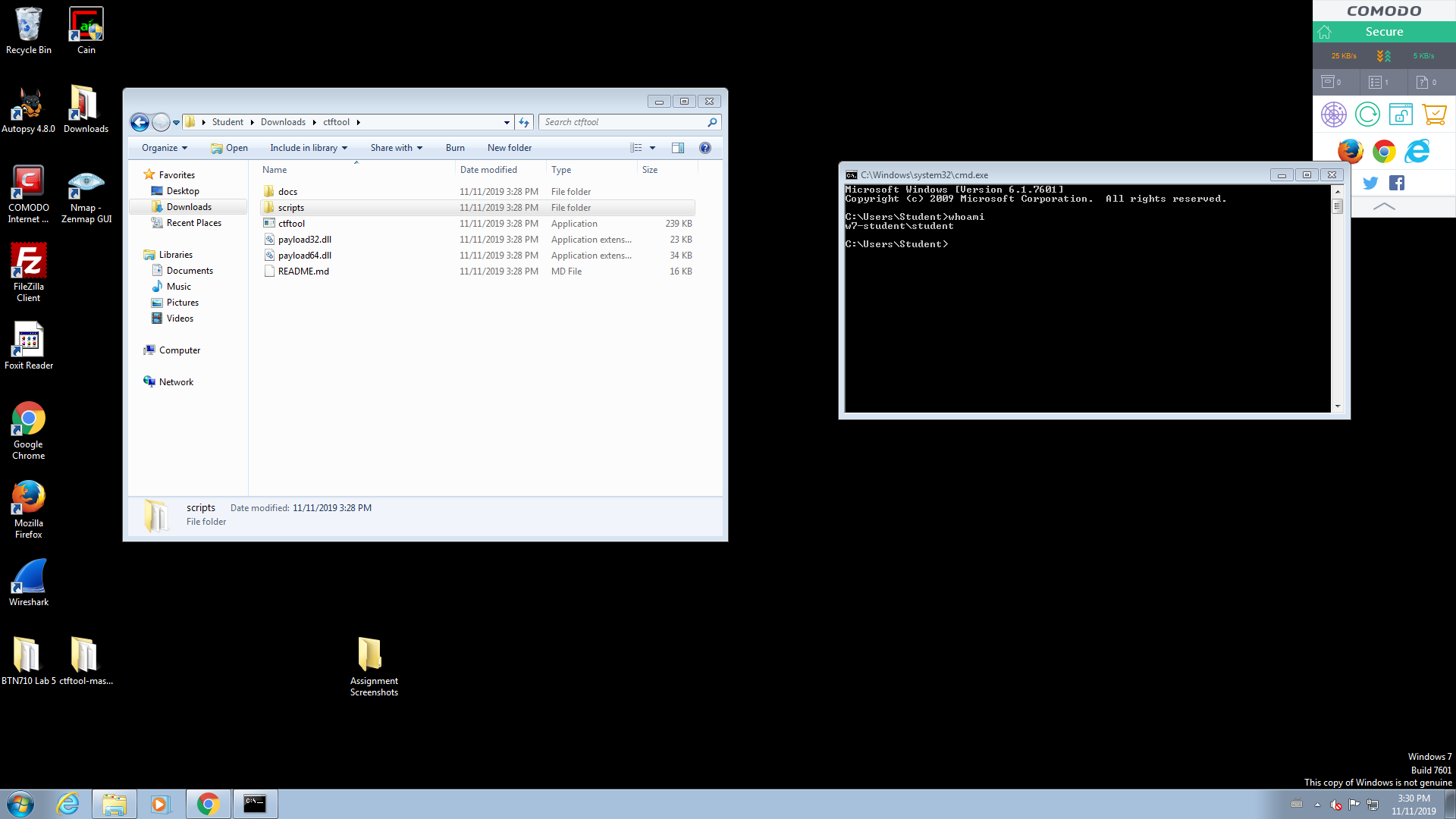
CTF protocol has no documentation referring to what the acronym stands for, however there is a speculation that it stands for “Common Text Framework”. Text Services Framework utilizes CTF protocol and is actually the specific part of the TSF that allows for the exploit in this report to exist. The CTF protocol is not secure and exposes the ability for hackers to exploit escalation of privilege for Windows operating systems starting from Windows XP.

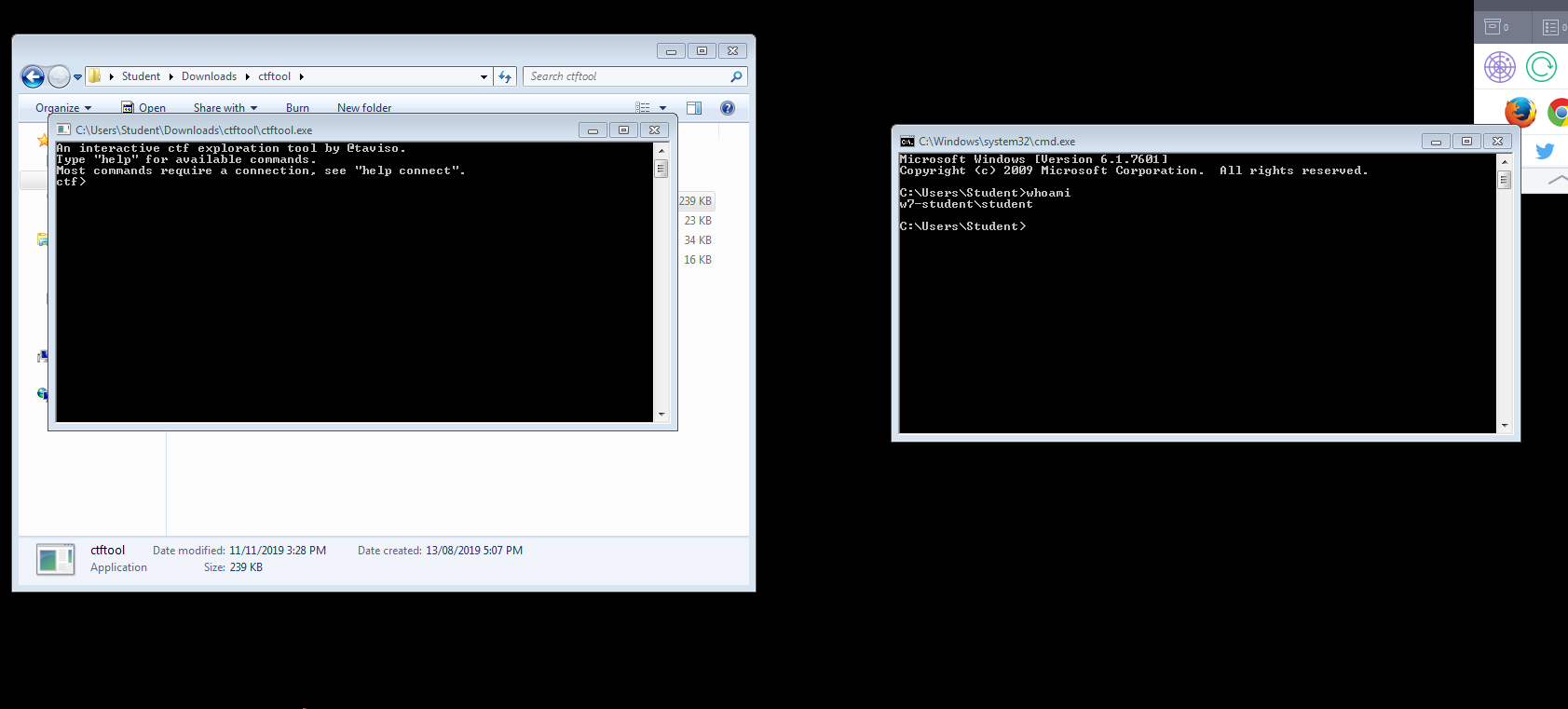
CTF Tool is an open-source command-line application that allows anyone to experiment with vulnerabilities in the CTF protocol within the TSF. It is readily available on GitHub and claims to be tested on and working with Windows 7, 8, and 10. We will be using and exploring this command-line application to exploit escalation of privileges on Windows through vulnerabilities in CTF protocol.

## How the Exploit Works and Description of the Attack

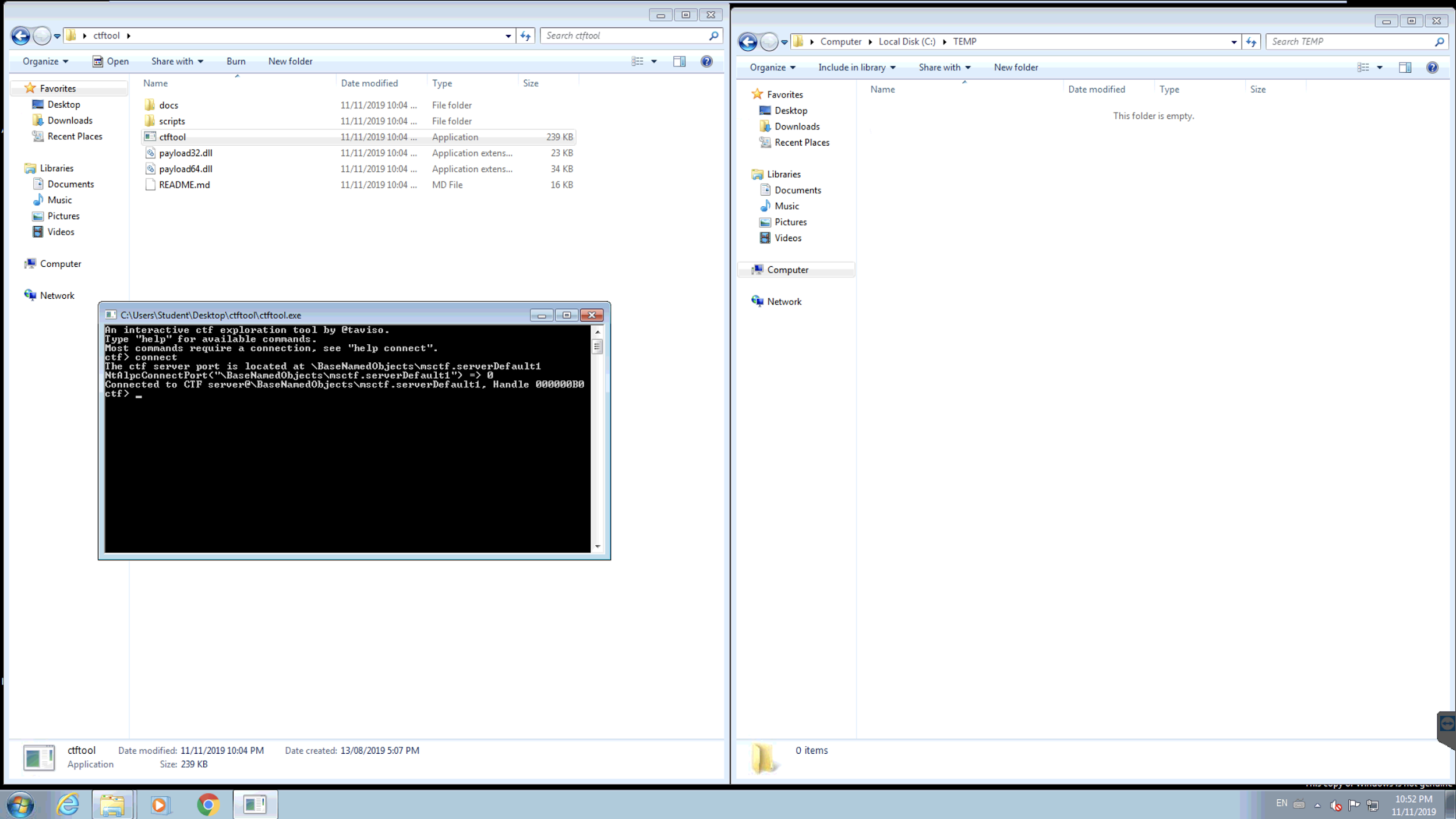
A close up of a white wall

Description automatically generated

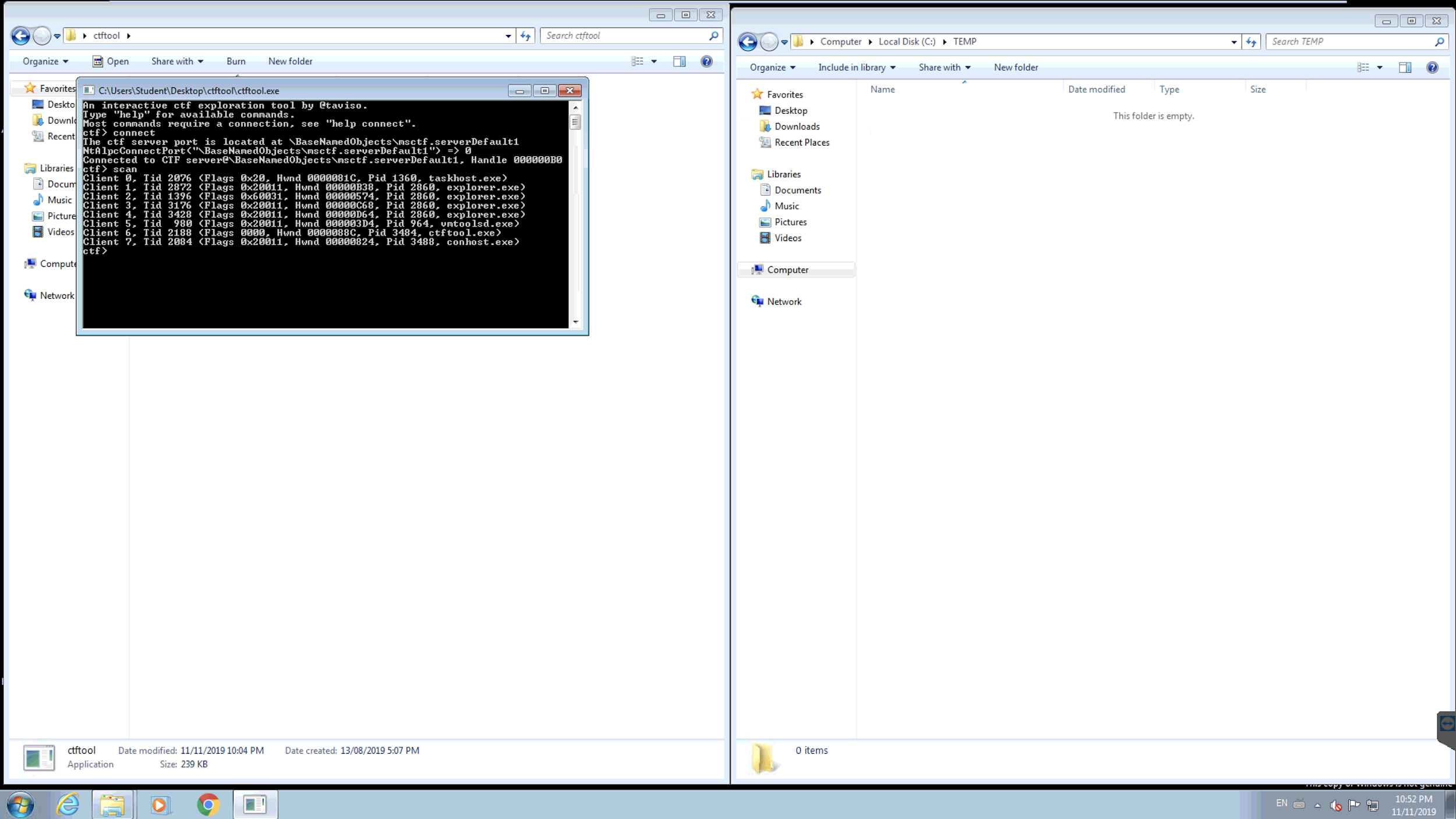
As explained previously in “Description and Diagram of Network” after all prerequisites are met on the system only then will the exploit work. This exploit can be run in different ways. However, for simplicity this document will only outline one easy-to-use method. Through the use of consent.exe this exploit can elevate the authority level of the user to full SYSTEM authority. When using CTF protocol, there is no access control, which causes a huge vulnerability. Any user can connect to a session of CTF without any authentication. An attacker can also pretend to be a CTF service, which allows even restricted applications to be accessed through this service. To begin a user must have the ctftool.exe installed. The user will run this program and begin entering commands. 



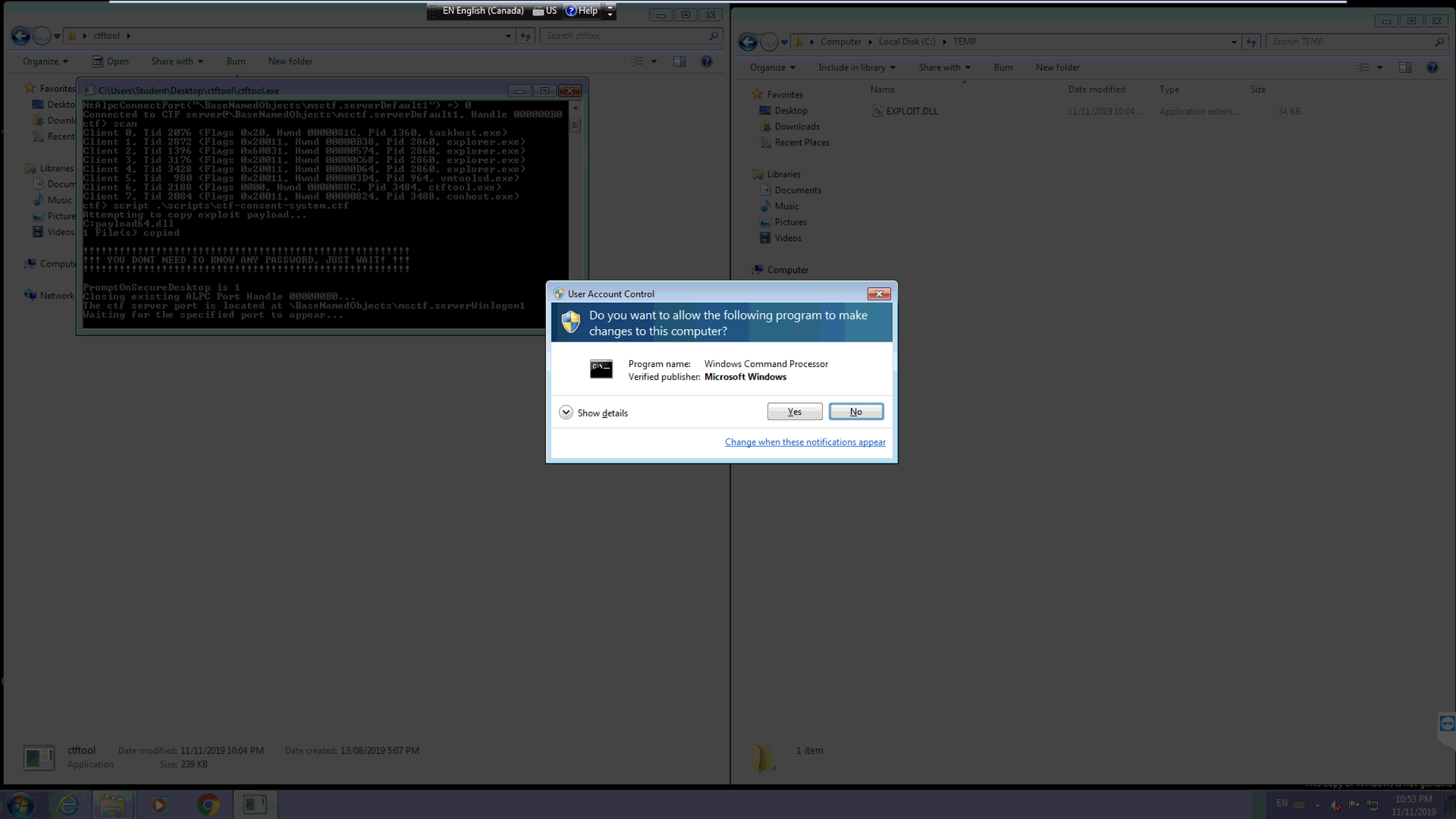
CTF Tool is used like a command prompt in the sense of how the user interacts with the program. The user will first type ‘connect’ which determines the location of the CTF server port and proceeds to connect to it.



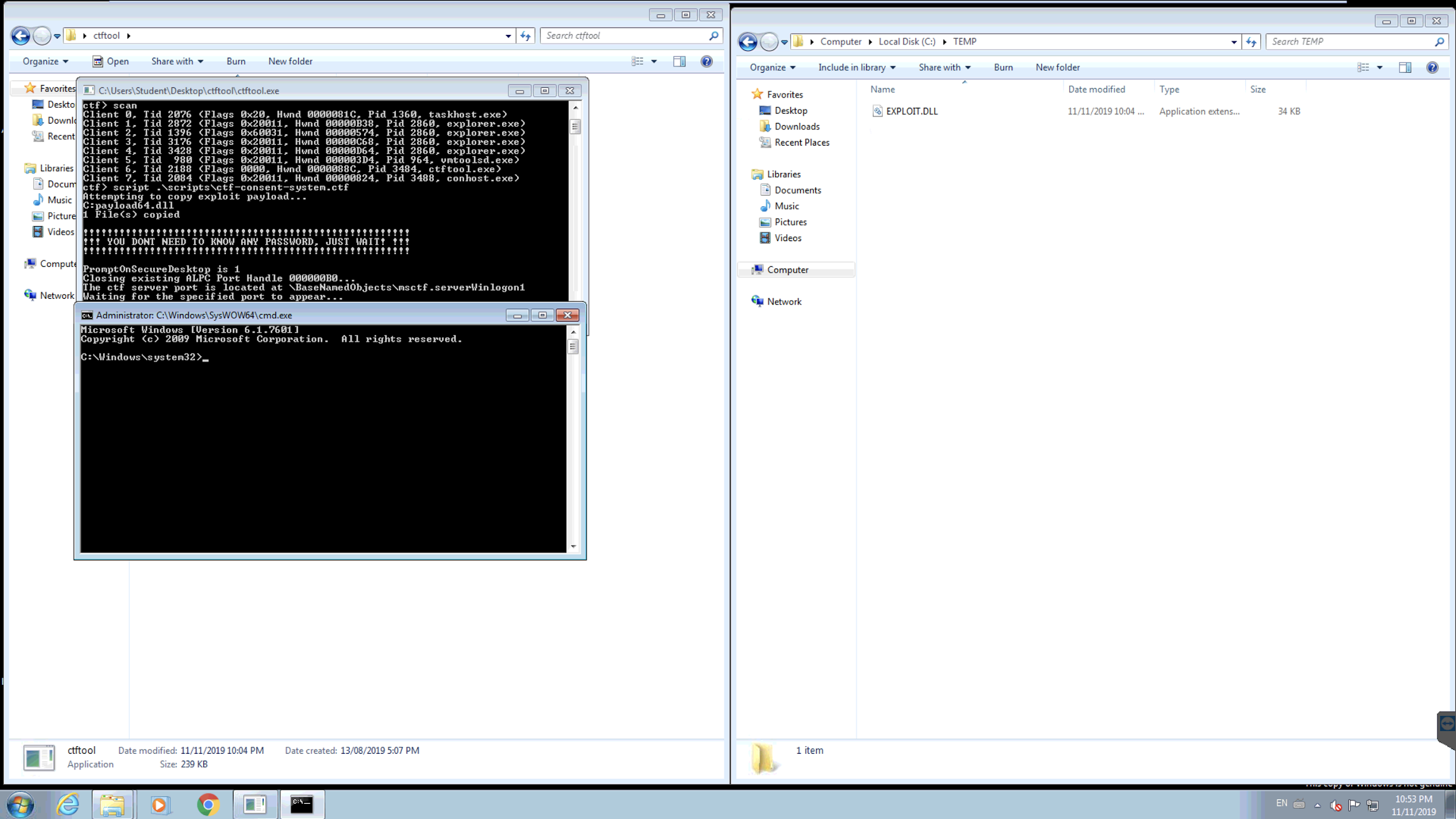
The next command entered will be ‘scan’, this command scans and displays all the active processes on the given server. Keep in mind that other functions like ‘wait’ or ‘lock’ can be used as well. By using the “wait” command, CTF Tool will wait until a client connects. For example, you can wait for notepad to be connected because it uses the CTF protocol. You can also connect to other user’s active sessions and take over their applications. This is very problematic when a user with admin privileges logs and you have access to applications that require admin privileges. Without even having an active admin session, by using “lock” command of CTF Tool, an attacker can gain admin privileges right away. Windows login interface uses CTF protocol, so it is compromised through the “lock” command and the attacker is taken to the login screen, where he already has system level privileges. But as mentioned before we will use the ‘scan’ command.

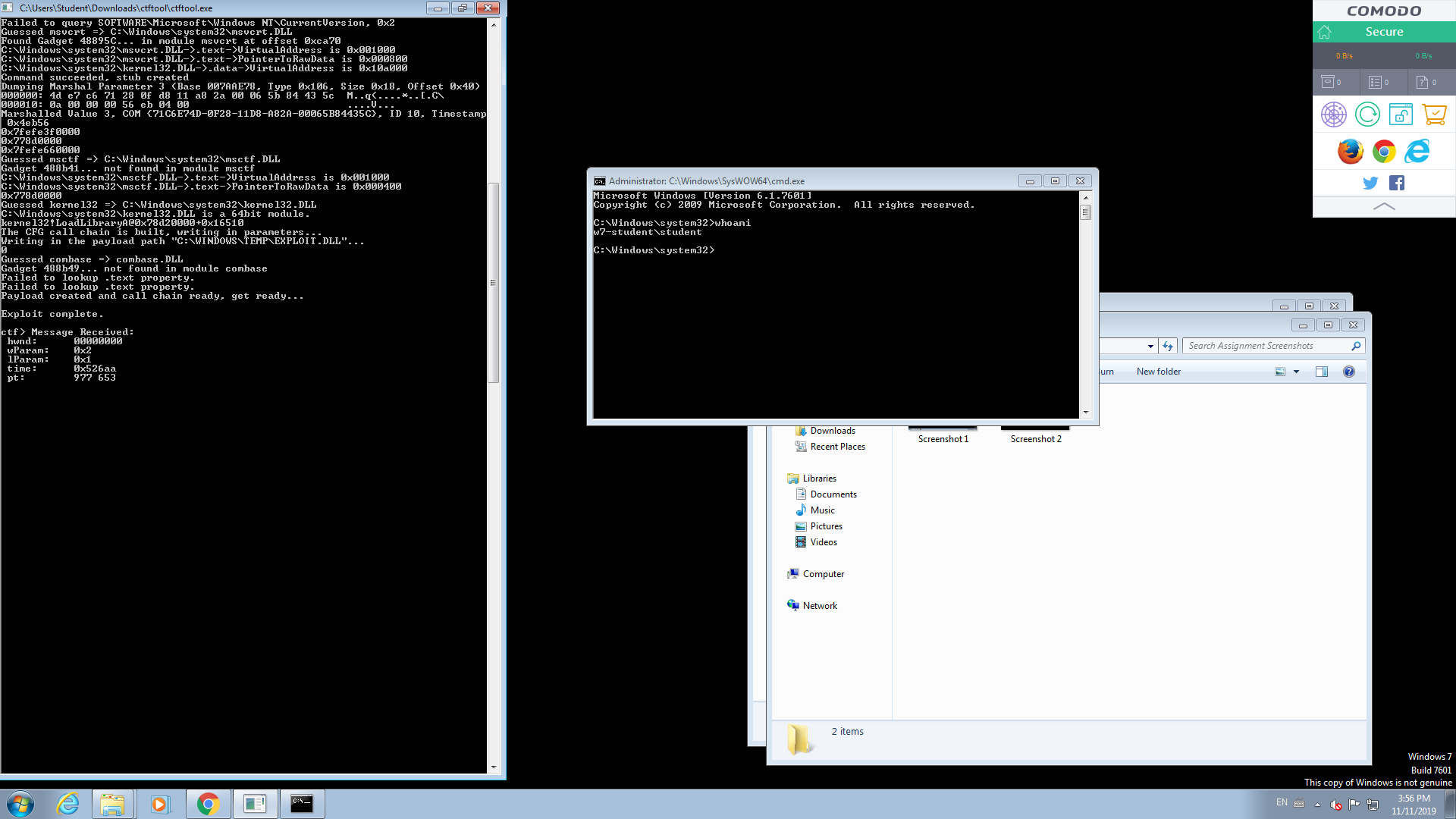


The next step to proceed with the exploit is to run the script associated with CTF Tool. In this particular example, “ctf-consent-system” is ran. This script triggers a user access control window to appear, once it appears CTF Tool hijacks that user access control and triggers code to launch an administrator command prompt window. While executing these commands the system will also leave a copy of a .dll file to a temp directory for it to run and hijack the system.



At this point, the payload is executed and runs the exploit. Once the payload has successfully ran another command prompt window will pop up, this is where the user can now check if the exploit was successful.

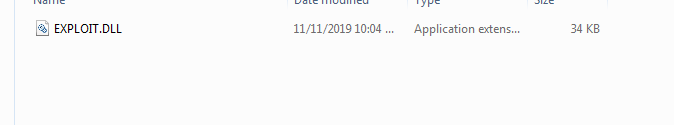




As the CTF protocol affects so many applications, any one of them can be exploited when they connect to a CTF session. An attacker can escalate their privileges, view what is happening on any client that is connected to the session, and even change. For example, the information inside the open notepad application can be viewed and edited since it is connected to the CTF session. An attacker can use this to read passwords as the user logs in, since the login screen also uses CTF. Any application that uses CTF can be compromised and there are a lot of entrance points for attackers to exploit.

## Signature of the Attack

For a successful exploit of the system a script must be run, whenever a script gets run by the CTF Tool, it always leaves a compiled “EXPLOIT.DLL” file in a temporary location as shown below.



If detected that DLL file can be read and edited with software like Visual Studio. When opened the DLL file can reveal functions or code that has been used by that particular file. To avoid .DLL script to be injected in a system, the easiest way to go about it would be to have an antivirus software or block access for DLL files to be added in the system.

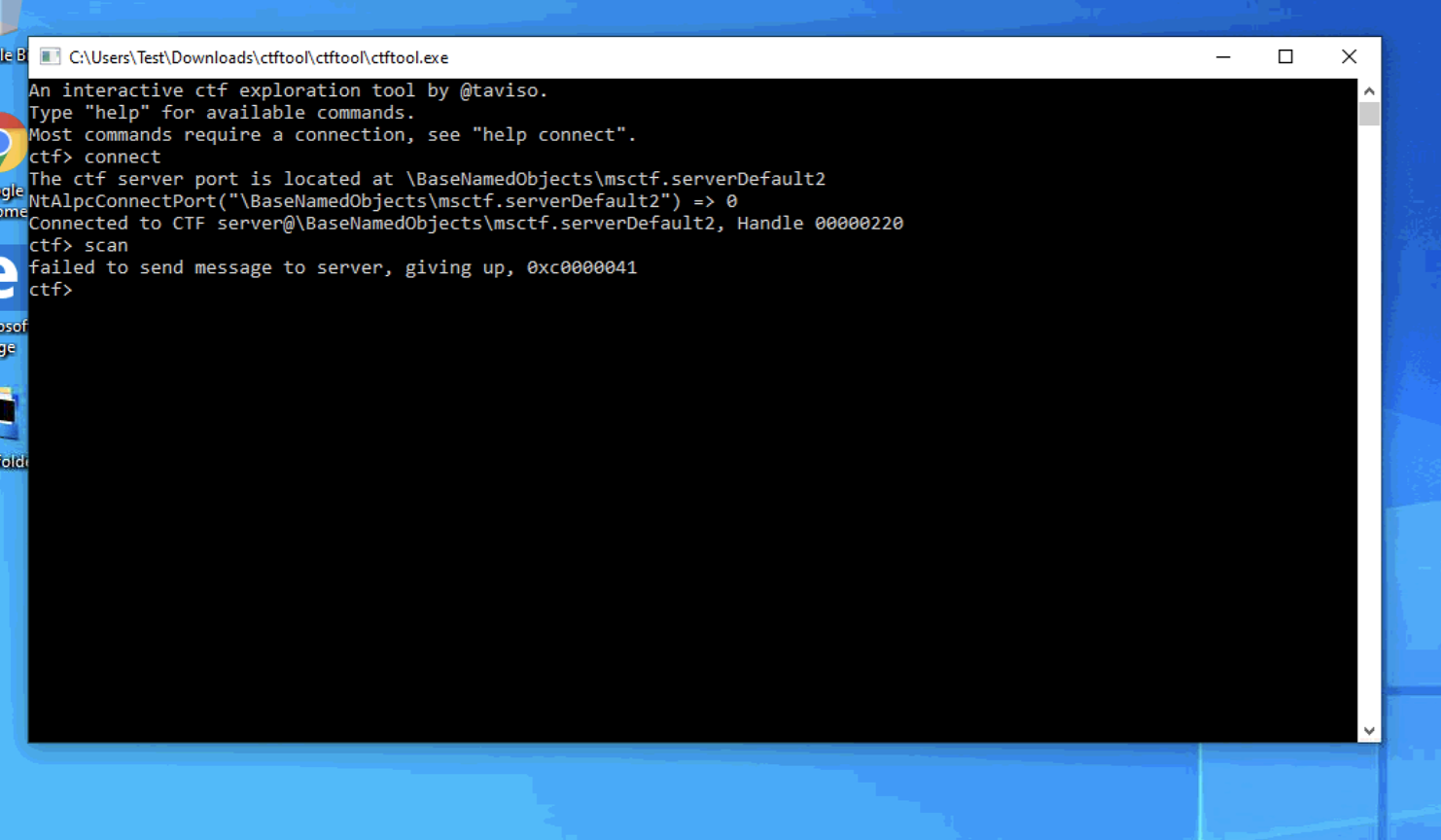
## How to Protect Against It

Even though there are no official protections against this exploit, there are some ways a user can mitigate the weaknesses through some simple, yet effective tasks. Although these tasks are effective, none of which are complete solutions to this exploit. If the program is developed to work around these mitigations all of which would prove to be non-effective.

1. Ensure the system is up to date with all security updates. There have been some updates as of August 2019 that have prevented the CTF tool from running the scan command which prevents the exploit. This however is not a permanent fix due to the fact that a payload can still be run manually.
2. Through the use of Windows Defender Realtime Protection.
3. Having an up to date AntiVirus software installed and running at all times. This would prove most effective on enterprise network computers due to the fact that it can be caught and prevented at that stage.
4. Use a native language that does not require the use of CTF, as of the newest software release from windows the only language that does not require CTF interaction is “English (United Kingdom)”.
5. Through the use of Windows Defender SmartScreen. This feature can be configured to prevent users from bypassing a warning and credential login which would prevent the attack by ctftool.exe. This however still does not solve the issue as it is just a mitigation.
6. A user or network can implicitly mitigate the login and consent pop up from launching child processes. Although this method is effective it does not leave any traces of a failed attempt. This means it would not be possible to detect if there was an attempt to exploit.
7. Download and install a mitigation service called HitmanPro. HitmanPro comes with a CTF guard allowing it to prevent the program from exploiting the CTF subsystem anytime it is being accessed.

## Remediated System Test

In August of 2019, Microsoft released a fix that would stop people from using CTF Tool, as seen below in the screenshot running windows 10. When trying to scan for active processes, although a user can connect to a server messages cannot be sent. This fix stopped attackers from being able to scan the network and also stopped attackers from being able to run script and exploit the system using CTF Tool.



# Section 3: Security Policy

This exploit is very hard to detect and prevent when using the affected versions of the Windows OS. However, the security policy can include rules that can protect users from these attacks. One of the rules that should be implemented with security policy that mandates users to update their system to the newest updates. Since the vulnerability inside TSF through CTF protocol became well known, Microsoft has been working to improve its security. They released a number of security patches over the last few months that addressed this issue. By keeping Windows up to date, one will gain access to these security patches that will protect oneself against this exploit. Although, this may sound very simple, this should be one of the main priorities of the security policy and will prevent unauthorized users from elevating their privileges. Another rule of security policy should mandate that users must install the most up to date antivirus programs that make it harder for this exploit to work. In addition, security policy should specify that employees keep their login information safe and not provide it to anybody else. On an enterprise network, an attacker still needs to gain access to a computer in the enterprise network to be able to escalate the privileges and gain access to other users. By preventing external users from having access to the enterprise network, the company will be able to keep their users safe. Ensure that everyone using the network is monitored to some degree and put flags in place that can notice the use of CTF Tool or a presence of an unauthorized user. By limiting access to the enterprise network, attackers will not be able to gain an entrance window to exploit the system.

Incidence reporting should include reports of this type of exploit. The damage to the company can be very severe if this exploit actually succeeds. The attacker will be able to gain access to pretty much everything within the enterprise network and have access to all the sensitive information. If this incident happens, it should be reported as an extremely severe breach. Enterprise network should be taken offline and new security patches must be installed on the computers that are missing them, so that this exploit cannot happen again.

# Conclusion

In conclusion, in the modern age of technological and software advancements allow for computer users to have nice features that simplify their life and allow software developers to create software much faster. However, new features and advancements may expose vulnerabilities to systems that haven’t existed before. For example, the Windows Text Services Framework simplifies the life for developers of all Windows applications that require user input. The developers of these applications don’t need to worry about which piece of hardware is collecting user input and just need to worry about the end product. In addition, the users of Windows enjoy cool features like translation of their voice into a written text through this framework. Although, TSF provides so many benefits and should not be removed, it also revealed a very dangerous vulnerability, where attackers can gain access to protected accounts and escalate their privileges on a Windows system.

Creators of software need to be mindful of the effects of their software and they need to prevent any possibility of attackers to exploit vulnerabilities within their systems. It seems disappointing that a vulnerability in the TSF through the CTF protocol existed since Windows XP, but only recently has it been noticed by Microsoft and patched. Seasoned hackers might have known about this exploit for years and kept it hidden. Only recently, when the exploit became public with the readily available CTF Tool that allows anyone to exploit this vulnerability, Microsoft realized how severe the situation is and quickly released some patches. Companies should always be mindful of exploits within their software and have dedicated teams that can find the vulnerabilities in their system before the hackers do. In addition, users need to be mindful of the software they use and security implications they provide. Users should also try to protect themselves by staying up to date on the newest software updates, which provide security patches and install antivirus and security programs that allow them to have another line of defense against hackers.

# References

<https://www.cvedetails.com/cve/CVE-2019-1235/>

<https://www.cybersecurity-help.cz/vdb/SB2019091033?affChecked=1>

<https://blog.stealthbits.com/using-ctftool-exe-to-escalate-privileges-by-leveraging-text-services-framework-and-mitigation-processes-and-steps/>

<https://www.tenable.com/plugins/nessus/128635>

<https://github.com/taviso/ctftool>

<https://googleprojectzero.blogspot.com/2019/08/down-rabbit-hole.html>

<https://docs.microsoft.com/en-us/windows/win32/tsf/new-features-in-tsf>

<https://www.zdnet.com/article/vulnerability-in-microsoft-ctf-protocol-goes-back-to-windows-xp/>

<https://news.sophos.com/en-us/2019/08/22/blocking-attacks-against-windows-ctf-vulnerabilities/>

<https://www.us-cert.gov/ncas/current-activity/2019/08/15/microsoft-releases-security-update-windows-elevation-privilege>

<https://www.partitionwizard.com/partitionmagic/vulnerability-in-microsoft-ctf-protocol-006.html>

<https://www.cvedetails.com/cve/CVE-2019-0735/>

<https://www.cvedetails.com/cve/CVE-2018-6674/>