

# Using LabVIEW To Do Totally Legal Things Online

Eric Stach

Duke University, Department of Civil and Environmental Engineering

July 25, 2024



Duke

# #OurGiantsAreFemale



- Dr Genevieve Lipp, Assistant Professor of the Practice in Mechanical Engineering and Electrical and Computer Engineering at Duke University
- Teaches classes including Dynamics, Robotics, Control Systems (uses LabVIEW 😊), programming for non-programmers
- Recently named Director of Duke's New First Year Computing program required by all freshman engineers beginning Fall 2024 (not LabVIEW 😞)
- Every Duke engineers' first meaningful college programming course will be taught by Genevieve

# By The End of This Session You Will:

- Identify basic HTTP Terminology
- Learn how to interact with the information on a web page through LabVIEW
- Learn About a FREE 3<sup>rd</sup> party tool that Accelerates Web Development
- NOT become Java or HTML Expert
- NOT Become a Hacker or Malicious Internet User (**PROMISE ME**)

# About Me



- BSE in Mechanical Engineering and Materials Science from Duke University
- Working at Duke since 2016
  - First in MEMS designing teaching lab experiments and writing LabVIEW code for test stands
  - Current: R&D Engineer in Civil and Environmental Engineering Department designing wireless embedded sensor systems
    - I'm an ME in CEE doing ECE...makes no sense
    - Work mentioned in this presentation funded by ONR N00014-23-S-B001

# NEW USER GROUP PLUG

- We've started a new LabVIEW User Group in Durham, NC!
- If you live in the Durham, Raleigh, Chapel Hill, or Research Triangle Park area, come join us!!
- Next Meeting is August 21, 2024

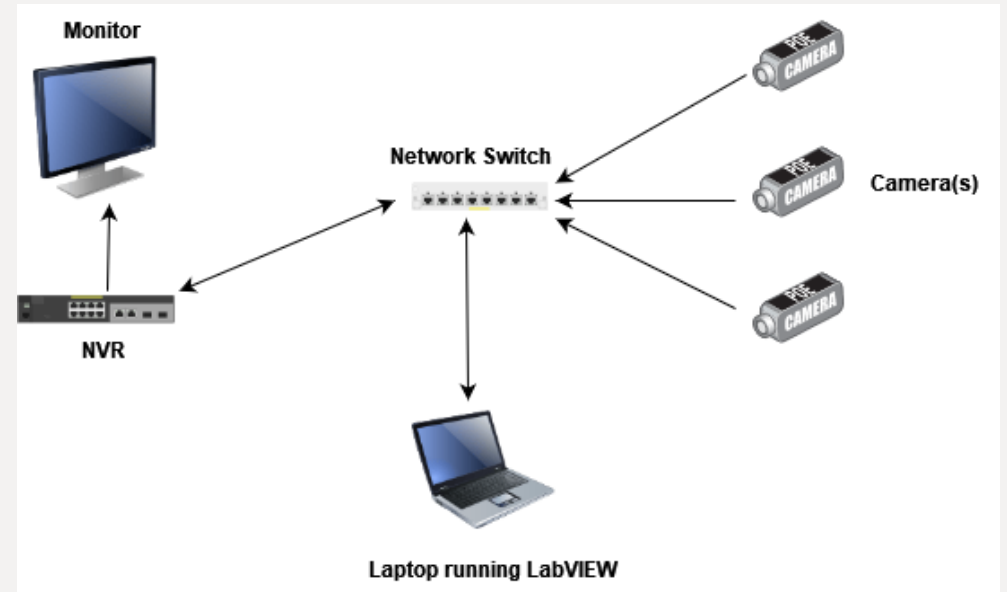


# Background

- Network video recorder (NVR) standalone device that controls and stores IP camera feeds
- NVR handles recording, storage, settings
- Port for mouse, monitor to use device
- Also has a network port (as suggested by the name)
- NI DAQ collecting analog measurements with LabVIEW and NVR for storing IP camera video
- Test runs involved starting each device separately (NVR required navigating multiple menu screens)
- Improve setup by integrating NVR start/stop record functionality into LabVIEW application
  - Less cumbersome, less equipment



**Figure 1.** Network Video Recorder



**Figure 2.** System Diagram

# Plan A

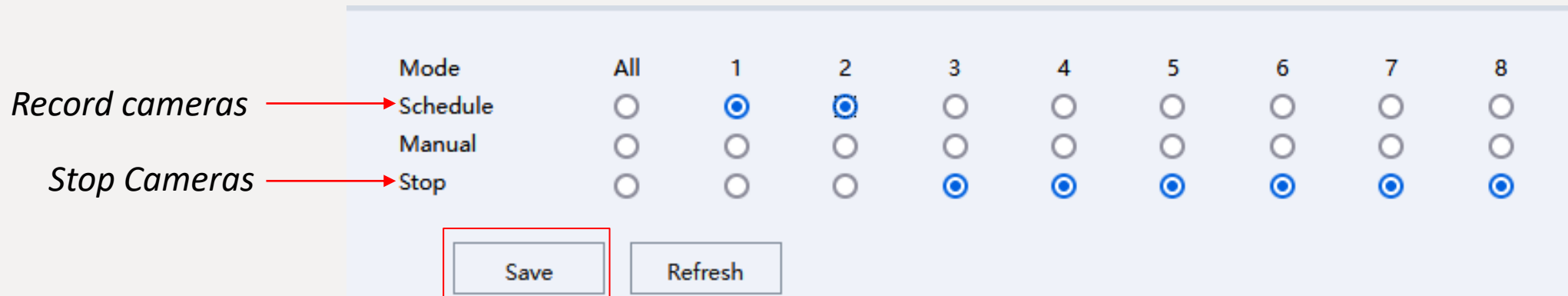
- Communicate via TCP/UDP
- I'll just peek at the manual...
- No documentation in user manual
- Reach out to company— any API?
  - Response:
    - Company: We have no API. The manufacturer does not provide for security reasons
    - Me: Can we try asking the manufacturer?
    - Company: We are not allowed to give their information
    - Me: ☹️



Not helpful

# Well, it has a web client? So...Plan B?

- There is a web client that allows you to use NVR thru web browser
- In LabVIEW, can I perform the action of changing the radio button values then trigger the action that results from pressing Save?



**Figure 3.** NVR Web Interface Start/Stop Recording Screen



# HTTP Terminology

- HTTP (Hypertext Transfer Protocol): application layer used in the internet
- POST: request method for a web server to accept data enclosed in the body of a message
  - Formatted like XML, potentially
  - Has a header, body
  - Often used when uploading file or submitting a form, click button, sending post request
- GET: retrieve information from the web server
  - Opening a webpage
- Must define a URL to send the request

**I determine I will need to make a POST request, but what should be the message? And URL?**

Source: [https://en.wikipedia.org/wiki/HTTP#Request\\_methods](https://en.wikipedia.org/wiki/HTTP#Request_methods)

# Web Pages – Under The Hood

- Right-click on a webpage (CTRL+U) → View Page Source
  - Source Code for the webpage
- Search Source Code for Web Client, Search for Relevant Terms...
  - Schedule is term used to start recording
- No hits...
- Stop not found either

```
173 </div>
174 <!--在线用户 暂时注释掉 系统下面只保留 版本信息, 基本设置, 时间日期, 日志信息 17.8.4-->
175 <div class="bsd" style="display: none;" onclick="Oline_user()">
176 <div class="Config_li_icon"></div>
177 <div class="config_system_box">
178 <div class="config_system_bicon"></div>
179 <label class="ml10" id="Oline_user">在线用户</label>
180 </div>
181 </div>
182 <!--录像状态 暂时注释掉 系统下面只保留 版本信息, 基本设置, 时间日期, 日志信息 17.8.4-->
183 <div class="bsd" id="TimeDate_div" onclick="System_localset()">
184 <div class="Config_li_icon"></div>
185 <div class="config_system_box">
186 <div class="config_system_bicon"></div>
187 <label class="ml10" id="TimeDate">时间日期</label>
188 </div>
189 </div>
190 <div class="bsd" id="SecurityManage_div" onclick="SecurityManageSet()">
191 <div class="Config_li_icon"></div>
192 <div class="config_system_box">
193 <div class="config_system_bicon"></div>
194 <label class="ml10" id="syssecurityManage"></label>
195 </div>
196 </div>
197 <div class="bsd" id="DisplaySet_div" onclick="DisplaySet()">
198 <div class="Config_li_icon"></div>
199 <div class="config_system_box">
200 <div class="config_system_bicon"></div>
201 <label class="ml10" id="sysdisplay"></label>
202 </div>
203 </div>
204 <div class="bsd" id="YuntaiControl_div">
205 <div class="Config_li_icon"></div>
206 <div class="config_system_box" onclick="PTZControlCon()">
207 <div class="config_system_bicon"></div>
208 <label class="ml10" id="sysYuntaiControl">云台控制</label>
209 </div>
```

schedule ^ v  Highlight All  Match Case  Match Diacritics  Whole Words  Phrase not found

Figure 4. Searching NVR Interface Code for 'Schedule' Keyword

# Web Pages – Search Web Client

- Oh, there's some Javascript links at the end!
- Specifically, one called *Video\_control.js*
- Super! Let's see what this yields...

```
624 <script type="text/javascript" src="js/setting/AlarmSetting.js" charset="utf-8"></script>
625 <script type="text/javascript" src="js/setting/VideoBlind.js" charset="utf-8"></script>
626 <script type="text/javascript" src="js/setting/VideoLoss.js" charset="utf-8"></script>
627 <script type="text/javascript" src="js/setting/AlarmEventException.js" charset="utf-8"></script>
628 <script type="text/javascript" src="js/setting/Configuration.js"></script>
629 <script type="text/javascript" src="js/setting/Encodset.js"></script>
630 <script type="text/javascript" src="js/setting/Wheelset.js"></script>
631 <script type="text/javascript" src="js/setting/Splitscreen.js"></script>
632 <script type="text/javascript" src="js/setting/Disk_basicset.js"></script>
633 <script type="text/javascript" src="js/setting/Disk_advancedset.js"></script>
634 <script type="text/javascript" src="js/setting/Disk_smartset.js"></script>
635 <script type="text/javascript" src="js/setting/Video_control.js"></script>
636 <script type="text/javascript" src="js/setting/User_account.js"></script>
637 <script type="text/javascript" src="js/setting/Automatic_maintenance.js"></script>
638 <!-- <script type="text/javascript" src="js/flowplayer-3.2.13.min.js"></script> -->
639 <script type="text/javascript" src="js/setting/eventLinkageComment.js"></script>
640 <script type="text/javascript" src="js/setting/LineCross.js"></script>
641 <script type="text/javascript" src="js/setting/AreaIntrusion.js"></script>
642 <script type="text/javascript" src="js/setting/UnattendedObject.js"></script>
643 <script type="text/javascript" src="js/setting/ObjectMissing.js"></script>
644 <script type="text/javascript" src="js/setting/RegionEntrance.js"></script>
645 <script type="text/javascript" src="js/setting/RegionExiting.js"></script>
646 <script type="text/javascript" src="js/setting/FastMoving.js"></script>
647 <script type="text/javascript" src="js/setting/outoutMode.js"></script>
```

Figure 5. Javascript code on NVR Interface



# I Don't Know What To Do With This Information

- Should I pack it in?
- Should I become an expert on web development for the convenience of a few button clicks?
- NONE OF THE ABOVE!
- I just need an *insight* (shoutout Norm) there's a tool that will give me one

# BurpSuite

- Application by PortSwigger for web application security testing
  - <https://portswigger.net/burp>
- Users include Amazon, NASA, Barclays
- Free community edition
  - Free edition has limited tools
- Many thanks to Sam Taggart for referral

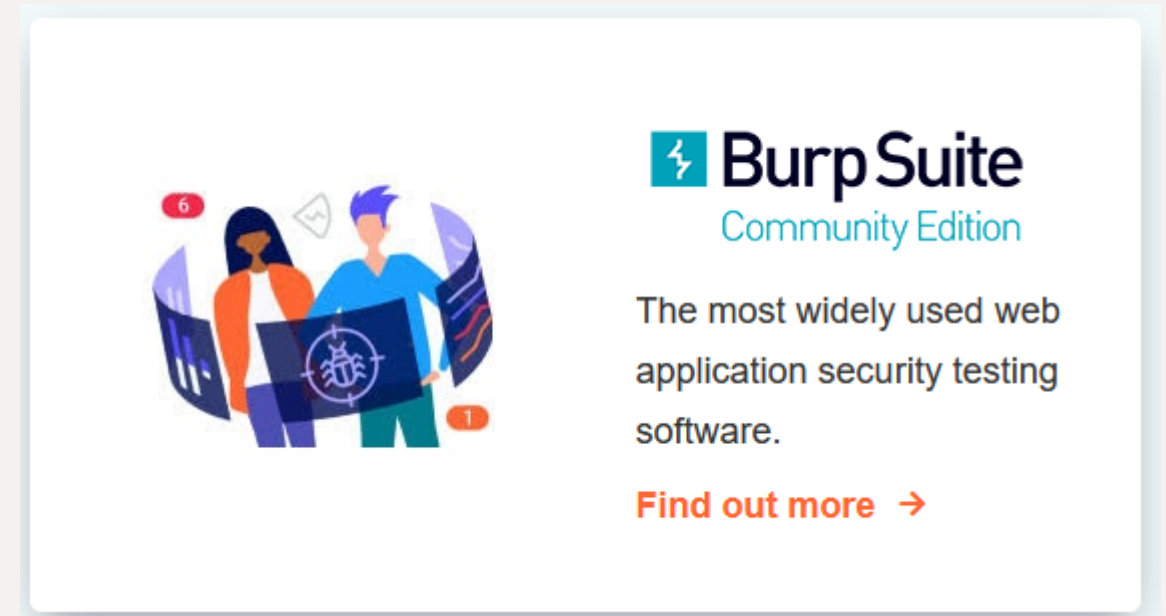


Figure 7. Helpful! But don't abuse it...

# Burp Suite – Proxy Intercept History

- Allows you to open web browser and intercept the HTTP requests to (among other things) view GET and POST history and URL
- Why is this useful? You can learn what is happening under the hood when you interact with a website
  - Also, a really great way to learn what 3<sup>rd</sup> parties (like FaceBook) are accessing your data on the webpage. A little freaky, but not surprising...
- Sometimes the URL is just the URL of the website, sometimes it's some URL that you never see

# Burp Suite – Intercept HTTP POST Request

The screenshot displays the Burp Suite interface with an intercepted HTTP POST request. The interface is divided into several sections:

- URL:** A table at the top shows the intercepted request details. The selected row (index 253) is highlighted with a red box. The table has columns for index, URL, method, path, status, size, and content type.
- Request:** A section on the left showing the raw HTTP request. It includes headers like Host, Content-Length, Accept-Language, User-Agent, Content-Type, Accept, Origin, Referer, Accept-Encoding, Cookie, and Connection. The body of the request is XML, starting with `<?xml version='1.0' encoding='utf-8'?><envelope><header><security>username</security><username>admin</username><password>888888</password></header><body><command>set.record.control</command><content><channel><id>1</id><record><mode>schedule</mode></record></channel><channel><id>2</id><record><mode>schedule</mode></record></channel><channel><id>3</id><record><mode>schedule</mode></record></channel><channel><id>4</id><record><mode>schedule</mode></record></channel><channel><id>5</id><record><mode>schedule</mode></record></channel><channel><id>6</id><record><mode>schedule</mode></record></channel><channel><id>7</id><record><mode>schedule</mode></record></channel><channel><id>8</id><record><mode>schedule</mode></record></channel></content></body></envelope>`. This section is also highlighted with a red box.
- Response:** A section on the right showing the raw HTTP response. It includes headers like HTTP/1.1 200 OK, Content-Type: application/xml; charset="UTF-8", Content-Length: 107, and Connection: close. The body of the response is XML, starting with `<?xml version="1.0" encoding="UTF-8"?><envelope><body><error>0</error><content></content></body></envelope>`. This section is also highlighted with a red box.

Annotations on the left side of the image point to these sections:

- POST Message Header:** Points to the header section of the request.
- POST Message Body:** Points to the XML body of the request.

At the bottom right of the response section, there is a text annotation: *Response – can parse to check if POST was successful*.

Figure 8. BurpSuite Interface when Pressing Save Button on NVR



# Generating the POST Request From LabVIEW

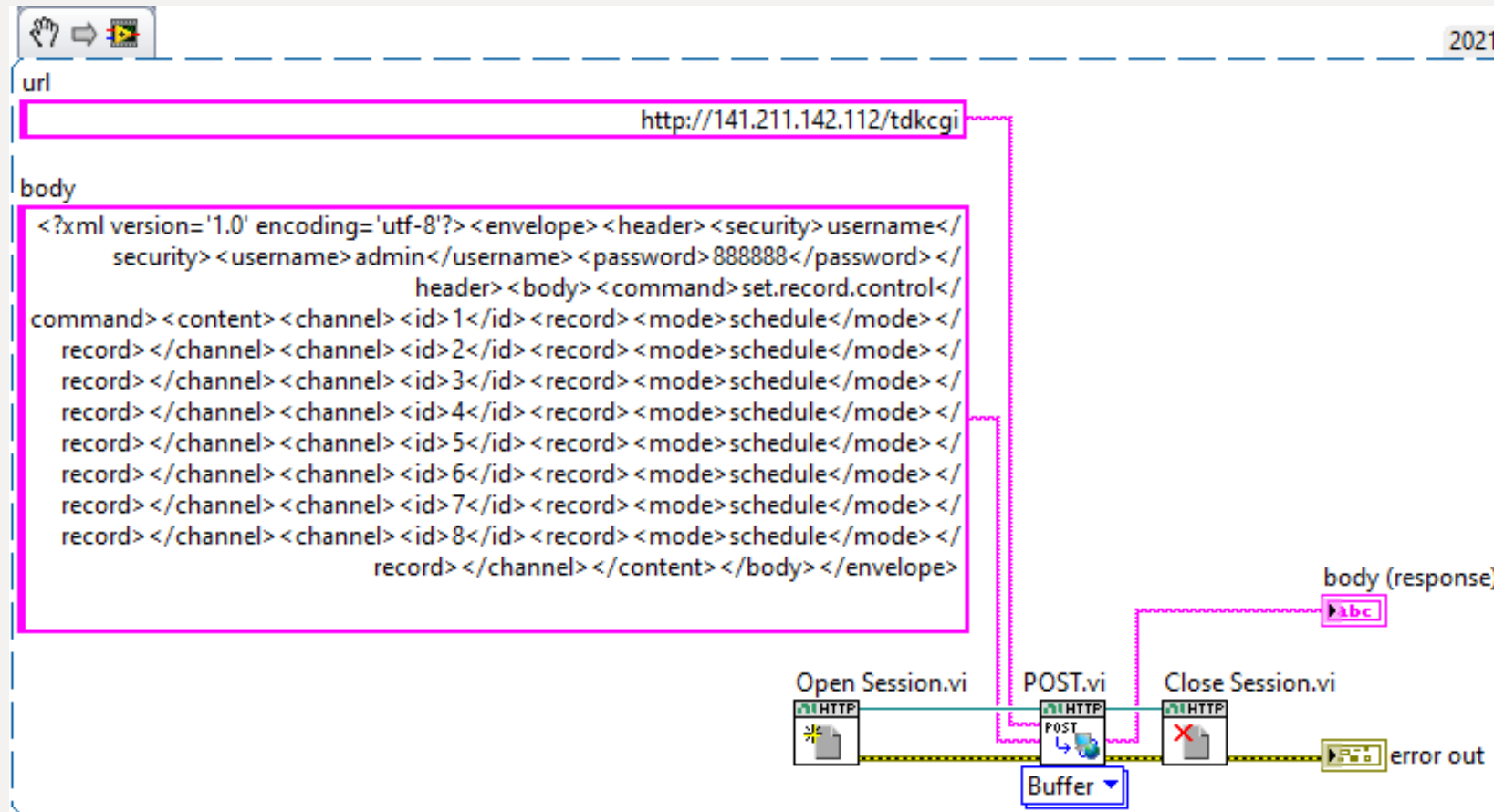
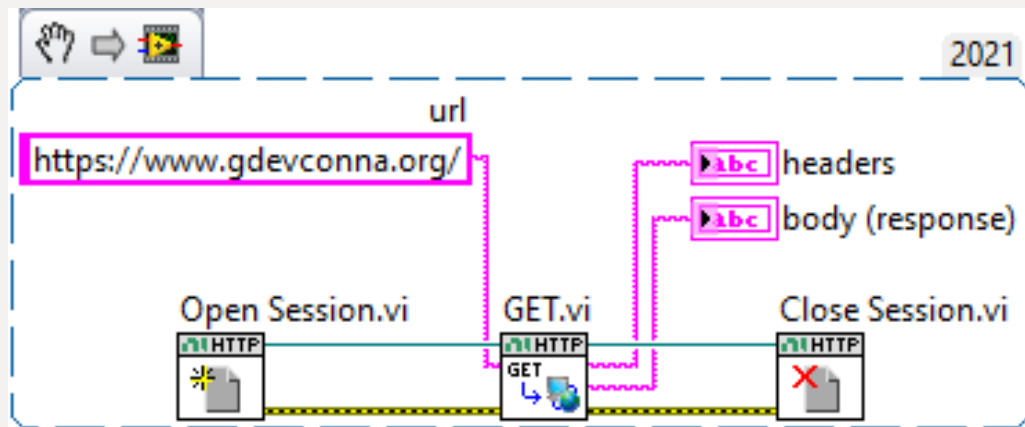


Figure 9. HTTP POST Request to Start Recording in LabVIEW

# Generating a GET Request From LabVIEW



Returns page source →

```
headers
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2024 03:37:18 GMT
Server: Apache
X-Content-Type-Options: nosniff
Content-language: en
X-UA-Compatible: IE=Edge
Link: </w/resources/assets/lvwiki.png?1fe0f>;rel=preload;as=image
Vary: Accept-Encoding, Cookie
Expires: Thu, 01 Jan 1970 00:00:00 GMT

body (response)
<!DOCTYPE html>
<html class="client-nojs" lang="en" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>LabVIEW Wiki</title>
<script> document.documentElement.className =
document.documentElement.className.replace(/^(\\s)client-nojs(\\s$)/, "$1client-js$2");</
script>
<script> (window.RLQ=window.RLQ||[]).push(function(){mw.config.set({"wgCanonicalName
space":"","wgCanonicalSpecialPageName":false,"wgNamespaceNumber":0,"wgPageName":
"Home","wgTitle":"Home","wgCurRevisionId":32268,"wgRevisionId":32268,"wgArticleId":258,
"wglArticle":true,"wglRedirect":false,"wgAction":"view","wgUserName":null,
"wgUserGroups":["*"],"wgCategories":["Main Page"],"wgBreakFrames":false,
"wgPageContentLanguage":"en","wgPageContentModel":"wikitext",
"wgSeparatorTransformTable":["",""],"wgDigitTransformTable":["",""],
"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","March","April",
"May","June","July","August","September","October","November","December"],
"wgMonthNamesShort":["","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct",
"Nov","Dec"],"wgRelevantPageName":"Home","wgRelevantArticleId":258,"wgRequestId":
"ZqB27sdI9IKgldyUO484TQAAAAE","wglProbablyEditable":false,
"wgRelevantPageProbablyEditable":false,"wgRestrictionEdit":"autoconfirmed"1
```

Figure 10. HTTP GET Request from Webpage

# An HTTP Based API For NVR (Take That, Anonymous Manufacturer!)

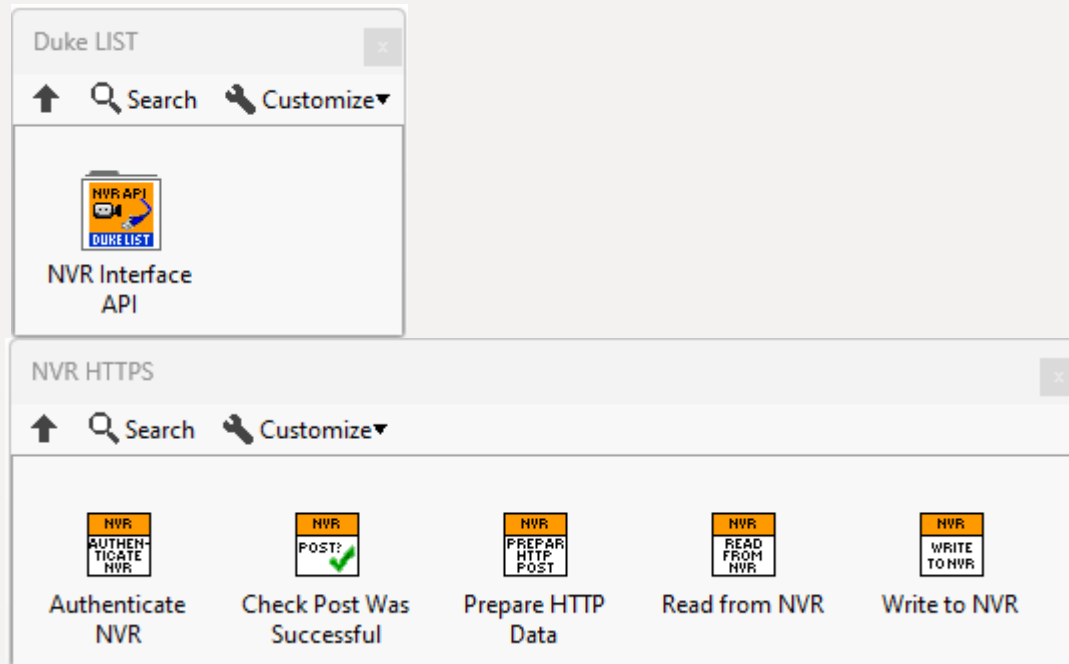


Figure 11. NVR Interface API Toolkit

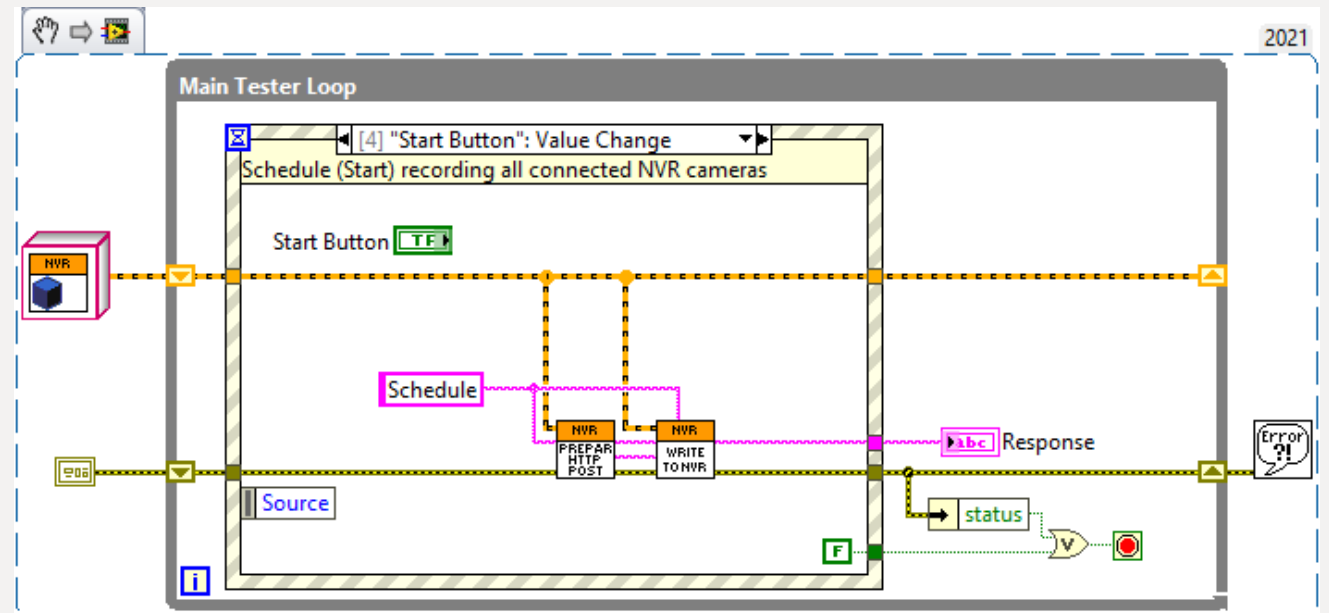


Figure 12. Toolkit Tester VI

# Demo #1: BurpSuite Demo

- Open Webpage, intercept with proxy, get HTTPS history

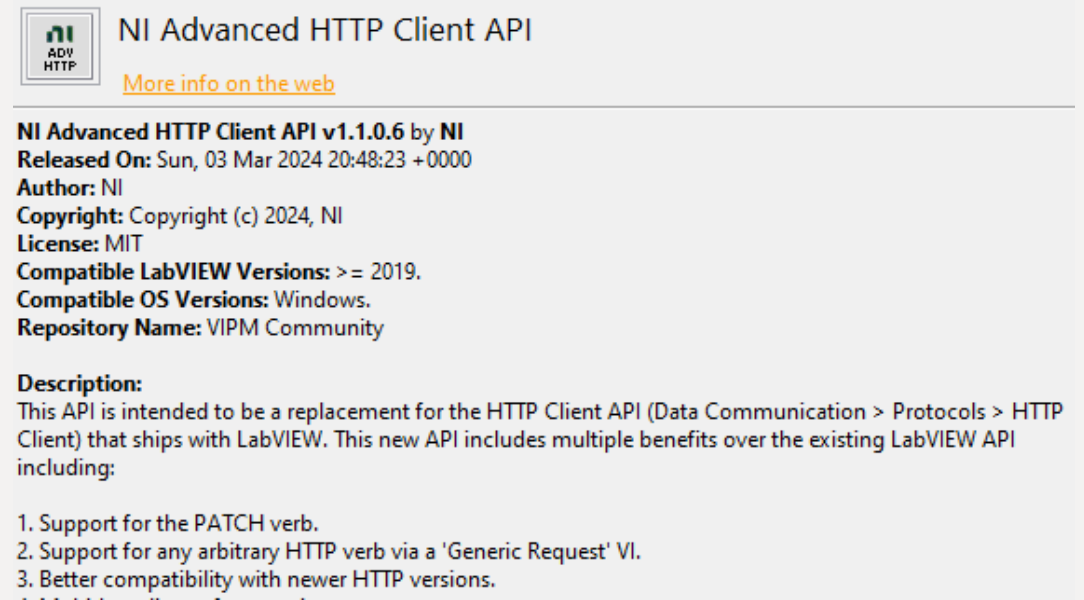
# Demo #2: (Theoretical) GDevCon NA Cookout

- It is summer, which means everyone is having a cookout
- Worse, my inbox is flooded with Google forms asking me to RSVP and signup to bring food
- I can't take it anymore so I'm going to create a LabVIEW tool to see what other attendees are bringing and to automate my responses based on what is needed

# HTTP In LabVIEW

- LabVIEW ships with HTTP functions (Connectivity → Protocols)
  - Don't use these
- I like the NI Advanced HTTP Client API (DNatt contributed so you know it will be readable)
  - Simple, easy to use, gets the job done
  - Intended be a replacement for the shipping API

**Figure 11.** VIPM Page for HTTP Client API

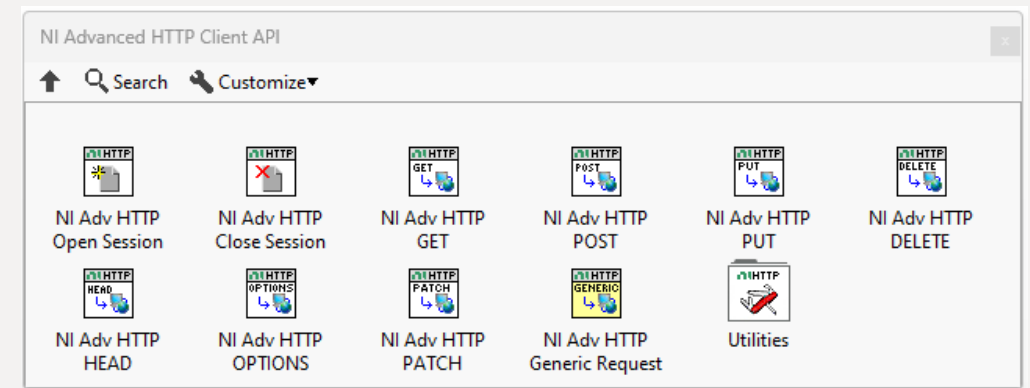


**NI Advanced HTTP Client API**  
[More info on the web](#)

**NI Advanced HTTP Client API v1.1.0.6** by NI  
**Released On:** Sun, 03 Mar 2024 20:48:23 +0000  
**Author:** NI  
**Copyright:** Copyright (c) 2024, NI  
**License:** MIT  
**Compatible LabVIEW Versions:** >= 2019.  
**Compatible OS Versions:** Windows.  
**Repository Name:** VIPM Community

**Description:**  
This API is intended to be a replacement for the HTTP Client API (Data Communication > Protocols > HTTP Client) that ships with LabVIEW. This new API includes multiple benefits over the existing LabVIEW API including:

1. Support for the PATCH verb.
2. Support for any arbitrary HTTP verb via a 'Generic Request' VI.
3. Better compatibility with newer HTTP versions.



**Figure 13.** HTTP Client API Palette

# WEB SECURITY (This Slide Included So Eric Reffett who talked about Security At GDevCon NA 2023 Doesn't Leave With His Head In His Hands)

- With enough time and energy, anything can be hacked
- Remove low hanging fruit
- Don't store passwords in VIs
- Messages could contain password information, remove passwords (separate lock from key)
- In my application, kept message template in database, with password removed, encrypted is good

# Use Your Newfound Web Skills Wisely...

- ~~Hacking!~~ BAD!
- ~~Malicious and reckless behavior!~~ BAD!
- Automating some boring stuff or adding some dynamic update capabilities from web or LAN sources to your LabVIEW application! PRETTY GOOD!



# Want to Learning More? Check Out These!

- Derrick's "Intro to NI Web HTTP & GET Dynamic Content"
  - [https://www.youtube.com/watch?v=iji\\_69tzcNM](https://www.youtube.com/watch?v=iji_69tzcNM)
- LabVIEW NTS "LabVIEW Code Check Internet Access (HTTP Method)"
  - <https://www.youtube.com/watch?v=edhLgSWN3og>
- NI Advanced HTTP API
  - [https://www.vipm.io/package/ni\\_lib\\_advanced\\_http\\_client\\_api/](https://www.vipm.io/package/ni_lib_advanced_http_client_api/)

MANY THANKS! Questions?

If the NSA asks, this presentation never happened