Supporting Information:
The Shape of and Solutions to the MTurk Quality Crisis

February 7, 2019

Table of Contents

A1 Model Details

Table A1, below, shows the results of OLS models predicting political knowledge scores and the logged time spent on the political knowledge questions in retrospective study 1.

Table A1. Political knowledge as a function of IP status

	Knowledge Score				Time on Knowledge			
VPS User	-0.28	*	-0.40	**	0.97	***	0.74	***
	(.14)		(.14)		(.10)		(.10)	
VPS Uncertain	0.28		0.30		0.33		0.34	
	(.37)		(.36)		(.26)		(.24)	
Foreign IP	0.04		0.02		0.78	***	0.55	***
	(.22)		(.22)		(.16)		(.15)	
Political interest	-		0.27	***	-		-0.03	
			(.05)				(.03)	
Education	-		0.03		-		0.03	
			(.04)				(.02)	
Male	-		0.17		-		-0.07	
			(.09)				(.06)	
White	-		-0.08		-		0.05	
			(.10)				(.07)	
Time on survey	-		-		-		0.56	***
							(.05)	
Constant	2.72	***	1.60	***	3.17	***	-0.16	
	(.05)		(.23)		(.04)		(.36)	
N	576		575		576		575	
R-squared	0.01		0.08		0.16		0.31	

Table A2, below, shows the results of OLS models predicting partisan identification as a function of political ideology, IP status, and interactions between each IP status and ideology. Valid respondents represent the omitted status condition.

Table A2. The relationship between ideology and partisan identification by IP status

	Partisan Identification			
	Study 1		Study 2	
VPS User	2.46	***	2.97	***
	(0.40)		(0.30)	
VPS Uncertain	-0.12		-1.28	
	(0.96)		(1.31)	
Foreign IP	1.37	*	1.60	***
	(0.60)		(0.41)	
Ideology	1.00	***	1.34	***
	(0.03)		(0.04)	
Ideology x VPS User	-0.45	***	-0.82	***
	(0.08)		(0.08)	
Ideology x VPS Uncertain	-0.04		0.43	
	(0.22)		(0.39)	
Ideology x Foreign IP	-0.46	**	-0.62	***
	(0.16)		(0.13)	
Constant	-0.08		-0.14	
	(0.13)		(0.11)	
N	576		1180	
R-squared	0.64		0.57	

Table A3. Experimental results by respondents IP status.

	Full Sample		Valid Sample		Fraudulent Sample	
Treatment 1	0.75	**	0.79	***	0.55	***
	(.07)		(.07)		(.17)	
Treatment 2	0.63	***	0.70	***	0.43	**
	(.06)		(.07)		(.15)	
Treatment 3	0.28	***	0.27	***	0.35	*
	(.06)		(.06)		(.16)	
Treatment 4	0.22	***	0.22	***	0.21	
	(.06)		(.06)		(.14)	
Treatment 5	0.11		0.04		0.49	**
	(.06)		(.06)		(.15)	
Treatment 6	0.03		-0.01		0.26	+
	(.06)		(.06)		(.15)	
Treatment 7	-0.11	*	-0.16	**	0.15	
	(.05)		(.06)		(.15)	
Treatment 8	-0.50	***	-0.60	***	-0.01	
	(.06)		(.06)		(.17)	
Treatment 9	-0.41	***	-0.54	***	0.25	
	(.06)		(.06)		(.18)	
Vignette 2	-0.03		-0.02		0.01	
	(.04)		(.04)		(.12)	
Vignette 3	0.04		0.04		0.06	
	(.04)		(.05)		(.11)	
Vignette 4	-0.08		-0.09	*	0.06	

	(.04)			(.04)			(.11)	
Vignette 5	0.11	**		0.10	*		0.21	
	(.04)			(.04)			(.12)	
Vignette 6	-0.13	**		-0.15	**		-0.01	
	(.04)			(.05)			(.12)	
Constant	-0.06			0.00			-0.37	**
	(.04)			(.05)			(.13)	
Respondents	576			480			96	
Observations	3456			2880			576	
R-squared	0.23			0.29			0.06	

A2 Comparison of data quality under different screening conditions

As was noted in the main paper, there were some differences between those who are labeled as potentially problematic by IP Hub, AbuseIPDB, and IPVOID. From this, the question naturally arises of how these differences affect the quality of the data that passes through these three screeners?

To see how IP Hub compares with these other services, we ran the first retrospective study through both IP Hub and the two services linked through Know Your IP (***27***). The results are in Table A4. As is clear, IP Hub produces similar results to AbuseIPDB, which is designed to track similar profiles. They agree on 97.3% of the cases. Conversely, IPVOID does not correspond with the results from IP Hub very well, agreeing on only 88.5% of cases. But, as noted in the main paper, this is likely because IPVOID's blacklist is not directly aimed towards VPS detection.

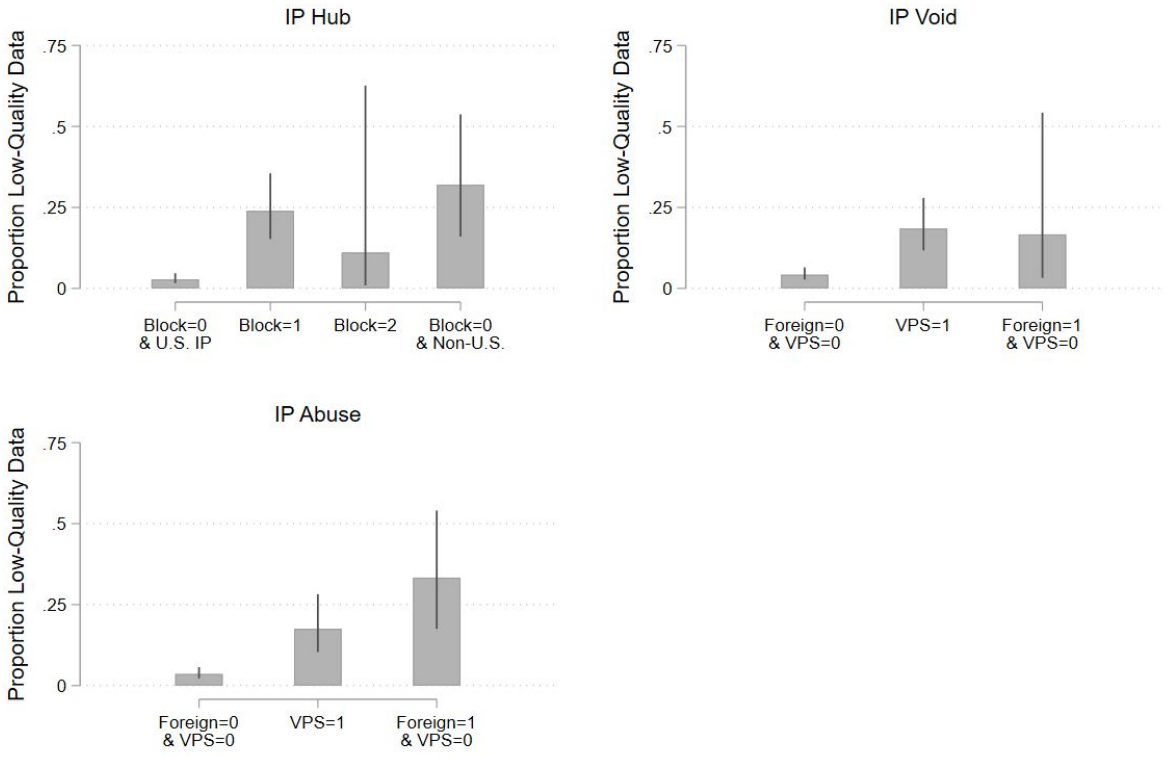
Table A4: Comparison Between IP Hub and Know Your IP

	AbuseIPDB Block	AbuseIPDB Safe	IPVOID Block	IPVOID Safe
IP Hub Block	91	5	67	29
IP Hub Safe	10	470	37	443

Nevertheless, as we show in Figure A1, those labeled as clean in all three datasets have approximately the same performance on our quality checks. As described in the manuscript, we included five data quality checks, such as whether respondents consistently reported their age and location and whether they could describe their task in the survey. We focus on a dichotomous indicator of whether a respondent was flagged as providing low-quality data according to at least one of the five measures. Following the analysis in the manuscript, we analyzed data quality among different IP types using each of the three alternative classifications schemes (IP Hub, AbuseIPDB, IPVOID).

The results are shown in Figure A1. As is clear, the three different IP services provide largely similar results. Respondents who were not flagged as fraudulent provided high-quality data across all three services (IP Hub: 2.8%, AbuseIPDB: 3.6%, IPVOID: 4.2%). All three services show higher rates of low-quality respondents among those who were flagged as outside of the U.S. (IP Hub: 32.0%, AbuseIPDB: 33.3%, IPVOID: 16.7%). All three services also show higher rates of low-quality respondents among those who are flagged for VPS use (IP Hub: 23.9%, AbuseIPDB: 17.6%, IPVOID: 18.5%). Overall, though the results are similar, IP Hub appears to be comparable in terms of finding true positives, while being more accurate in locating true negatives, which is useful for practitioners in avoiding dealing with complaints from their workforce.

Figure A1: Data quality on retrospective study 1 using different IP traceback services



A3 R and Stata packages for interaction with IP Hub

While this R package is already available on CRAN and, in a more updated form, on GitHub, for double-blind peer-review we have copied the main part of the package into this appendix. This is the main function, not including the other parts that are required for R packages.

Figure A2: Snapshot of R package code for IP traceback

```
#' Passes an array of IP addresses to iphub.info and returns a dataframe with details of IP
#'
#' Makes a call to an IP address verification service (iphub.info) that returns the information on the IP address, including the internet service provider (ISP) and whether it is likely a server farm being used to disguise a respondent's location.
#'@usage getIPinfo(d, "i", "key")
#'@param d Data frame where IP addresses are stored
#'@param i Name of the vector in data frame, d, corresponding to IP addresses in quotation marks
#'@param key User's X-key in quotation marks
#'@details Takes an array of IPs and the user's X-Key, and passes these to iphub.info. Returns a dataframe with the IP address (used for merging), country code, country name, asn, isp, block, and hostname.
#'@return ipDF A dataframe with the IP address, country code, country name, asn, isp, block, and hostname.
#'@note Users must have an active iphub.info account with a valid X-key.
#'@examples
#'id <- c(1,2,3,4) # fake respondent id's
#'ips <- c(123.232, 213.435, 234.764, 543.765) # fake ips
#'data <- data.frame(id,ips)
#'getIPinfo(data, "ips", "MzI3NDpJcVJKSTdIdXpQSUJLQVhZY1RvRxaXFsFW3jS3xcQ")
#'@export
getIPinfo <- function(d, i, key){
  if (!requireNamespace("httr", quietly = TRUE)) {
    stop("Package \"httr\" needed for this function to work. Please install it.",
         call. = FALSE)
  }
  if (!requireNamespace("utils", quietly = TRUE)) {
    stop("Package \"utils\" needed for this function to work. Please install it.",
         call. = FALSE)
  }
  if (!requireNamespace("iptools", quietly = TRUE)) {
    stop("Package \"iptools\" needed for this function to work. Please install it.",
         call. = FALSE)
  }
  #message("* Consider storing the ipDF as an object to write as an external df, e.g., write.csv(ipDF, 'ipDF.csv')")
```

```
  ips <- unique(d[ ,i])
  options(stringsAsFactors = FALSE)
  url <- "http://v2.api.iphub.info/ip/"
  pb <- utils::txtProgressBar(min = 0, max = length(ips), style = 3)
  ipDF <- c()
  for (i in 1:length(ips)) {
    if(is.na(iptools::ip_classify(ips[i])) | iptools::ip_classify(ips[i]) == "invalid") {
      warning(paste0("Warning: An invalid or missing IP address was detected on line ", i, ".
Please check this."))
      next
    }
    ipInfo <- httr::GET(paste0(url, ips[i]), httr::add_headers(`X-Key` = key))
    infoVector <- unlist(httr::content(ipInfo))
    ipDF <- rbind(ipDF, infoVector)
    utils::setTxtProgressBar(pb, i)
  }
  close(pb)
  rownames(ipDF) <- NULL
  ipDF <- data.frame(ipDF)

  return(ipDF)
}
```

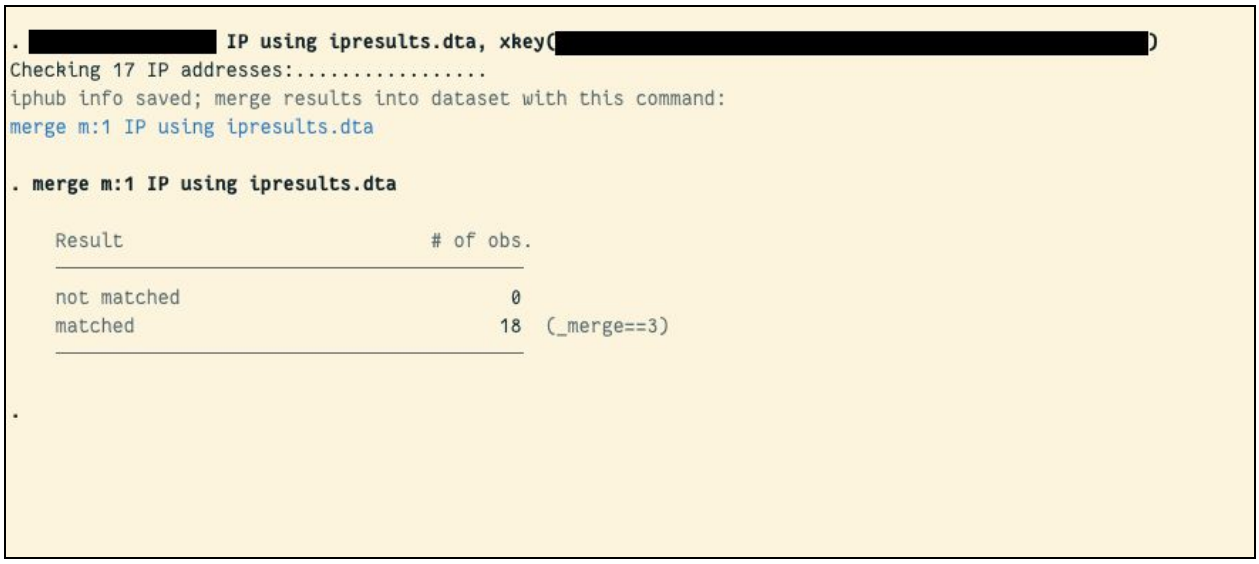
Figure A3 shows a call to the R package. It takes as its input a data frame, the name of the variable containing the IP addresses, and the user's IP Hub key. The output is also a data frame, which includes the IP address. A user can use the data frame independently or merge it with their data by the IP address.

Figure A3: An example of the R package run in R terminal

```
> library(rIP)
> id <- c(1,2,3,4) # fake respondent id's
> ips <- c("196.19.158.224", "199.241.146.238", "190.124.31.51", "181.225.47.154") # fraudulent ips
> data <- data.frame(id, ips)
> getIPinfo(data, "ips", "MzI2MTpkOVpld3pZTVg1VmdTV3ZPenpzMmhodkJmdEpIMkRMZQ==")
  |======================================================================| 100%
              ip countryCode                       countryName   asn
1  196.19.158.224          US                     United States 19969
2 199.241.146.238          US                     United States 25780
3   190.124.31.51          VE Venezuela, Bolivarian Republic of 61461
4  181.225.47.154          VE Venezuela, Bolivarian Republic of  8053
                  isp block        hostname
1      JOESDATACENTER     1  196.19.158.224
2 HUGESERVER-NETWORKS     1 199.241.146.238
3              Airtek     0   190.124.31.51
4                 IFX     0  181.225.47.154
```

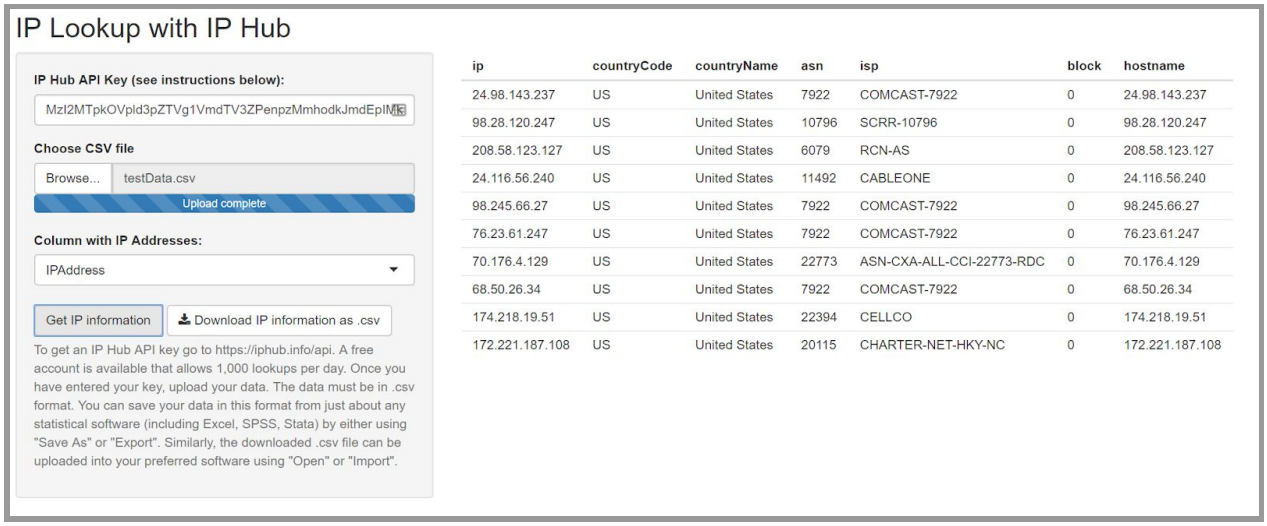
The Stata version of this package is available for download through SSC and can be installed by entering ***ssc install <nameofpackage>*** into the terminal. Running the program consists of a single command line. Figure A4 shows the simple call to the package and re-merging with the user's data.

Figure A4: An Example of the Stata Package Run in the Stata Terminal



Finally, some potential users of this package may be unfamiliar with R or Stata and, therefore, unable to use the package for their purposes. We created a Shiny application that can be used by anyone, regardless of software (e.g. SPSS, Stata, SAS, and Excel). The user can save their data as a .csv file using the "Save As" or "Export" feature in their software. The application takes that dataset and the user's key, and prompts them to enter the variable containing the IP addresses. When they press "Get IP Information", the R package is called and a data frame with the IP information is displayed in the main panel. Finally, the user can click the button to download the data, as a .csv file, to their computer and can merge it with their dataset in any software. The online application is available for use at <URL removed for peer review>.

Figure A5: Online application for IP lookups

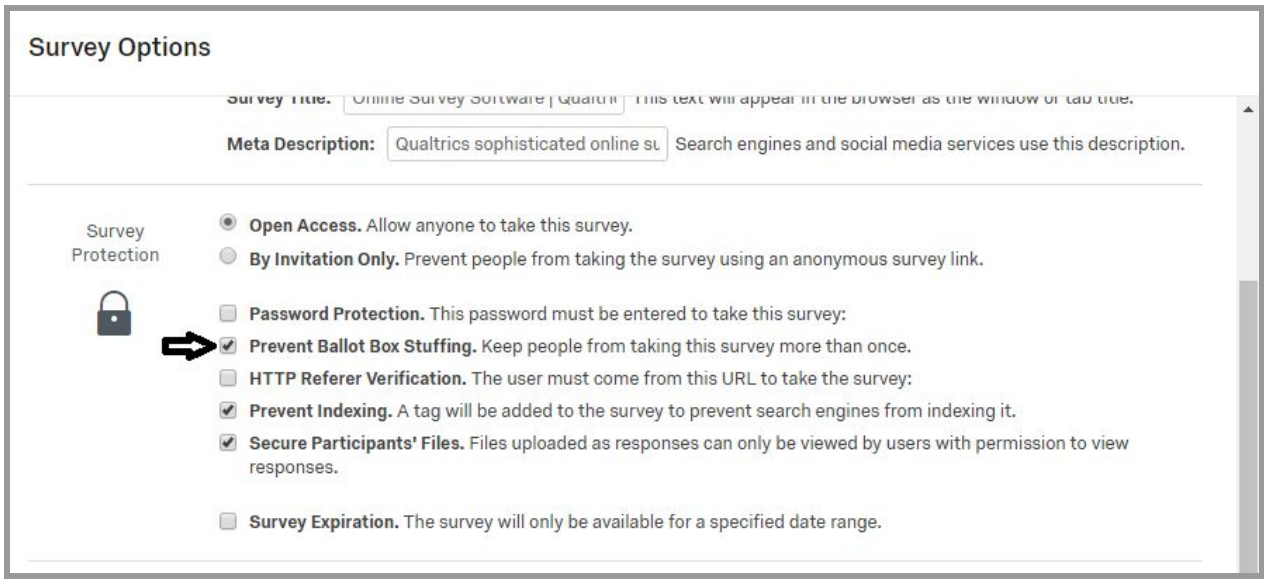


ip	countryCode	countryName	asn	isp	block	hostname
24.98.143.237	US	United States	7922	COMCAST-7922	0	24.98.143.237
98.28.120.247	US	United States	10796	SCRR-10796	0	98.28.120.247
208.58.123.127	US	United States	6079	RCN-AS	0	208.58.123.127
24.116.56.240	US	United States	11492	CABLEONE	0	24.116.56.240
98.245.66.27	US	United States	7922	COMCAST-7922	0	98.245.66.27
76.23.61.247	US	United States	7922	COMCAST-7922	0	76.23.61.247
70.176.4.129	US	United States	22773	ASN-CXA-ALL-CCI-22773-RDC	0	70.176.4.129
68.50.26.34	US	United States	7922	COMCAST-7922	0	68.50.26.34
174.218.19.51	US	United States	22394	CELLCO	0	174.218.19.51
172.221.187.108	US	United States	20115	CHARTER-NET-HKY-NC	0	172.221.187.108

A4 Qualtrics screening protocol for blocking VPS and international respondents

This protocol takes you through the steps of setting up a filter on Qualtrics that will block most people in a non-US location[1] or using a Virtual Private Server (VPS) to cover their location. It will take you through all the steps, with illustrations as needed.

Before you begin, if you do not want people to be able to answer your survey more than once, you should always enable the Prevent Ballot Stuffing option in Survey Options.



This option places a cookie on the user's browser to prevent them from answering the survey more than once from the same browser. It does not completely prevent duplicate responses, since users can take steps to erase or avoid detection through cookies. But it is a useful supplement to MTurk's built in checks to avoid duplication.

[1] This can also be used to select only participants from other countries by simply changing the IP_countryCode parameter.

Once this is done, you can follow the following steps to detect international and VPS respondents. The main concept is to lookup the IP address using a security service and use that information to make decisions on how to handle the potential respondent.

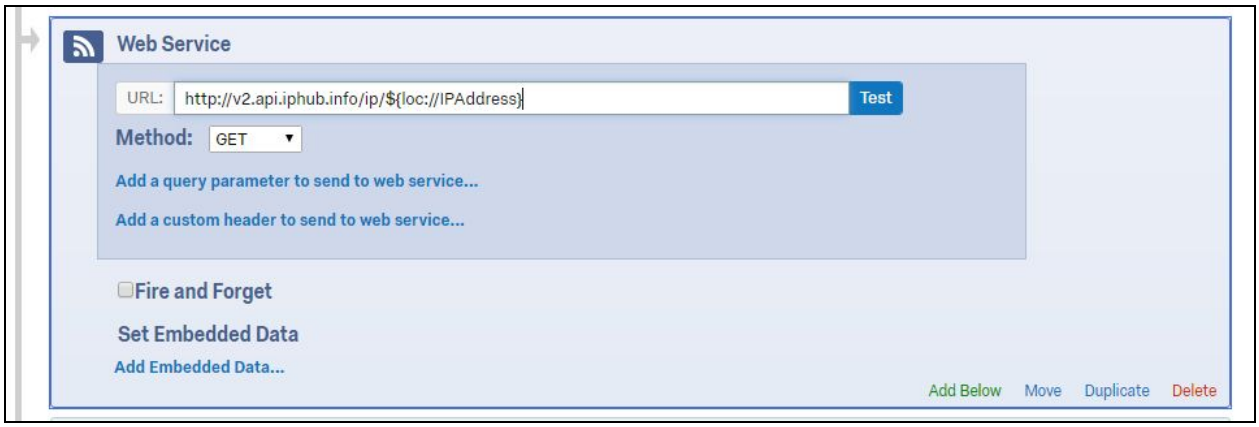
1. Create an account on IP Hub (https://iphub.info/pricing). A free plan that allows for 1,000 requests per day should suffice for most research purposes, although larger plans are available. We recommend IP Hub based on our own experiences, and because it provides a relatively liberal free service that functions quickly. You will be given an API key that consists of about 50 random letters and numbers, looking something like this:[2] MxI5ODpZT2kmVnlsR5iMcjBrRWpjxVZOKXIRKU1sNmdZb30EMA==

2. Next, you will want to be sure to add a warning to the beginning of the survey to tell people who are in the U.S. to turn off their VPNs or any ad blocking software they are using. This should be placed in its own block and should come before any other parts of the survey. This will prevent you from receiving complaints from some Turkers. From our piloting, it also appears that this is an effective way to initially screen out people who you do not want to take the survey (we noticed a significant drop in the number of international IPs testing our system once we added the warning).

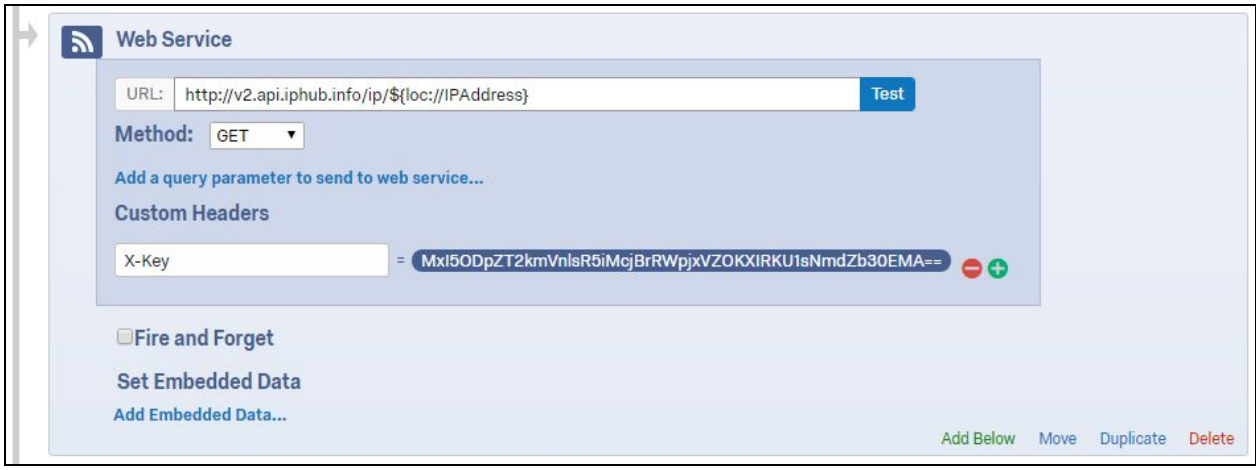
[2] This is not a real key, please do not try to use.

3. Next, go to the Survey Flow of your Qualtrics survey. After the block that contains the warning, you should add a Web Service.

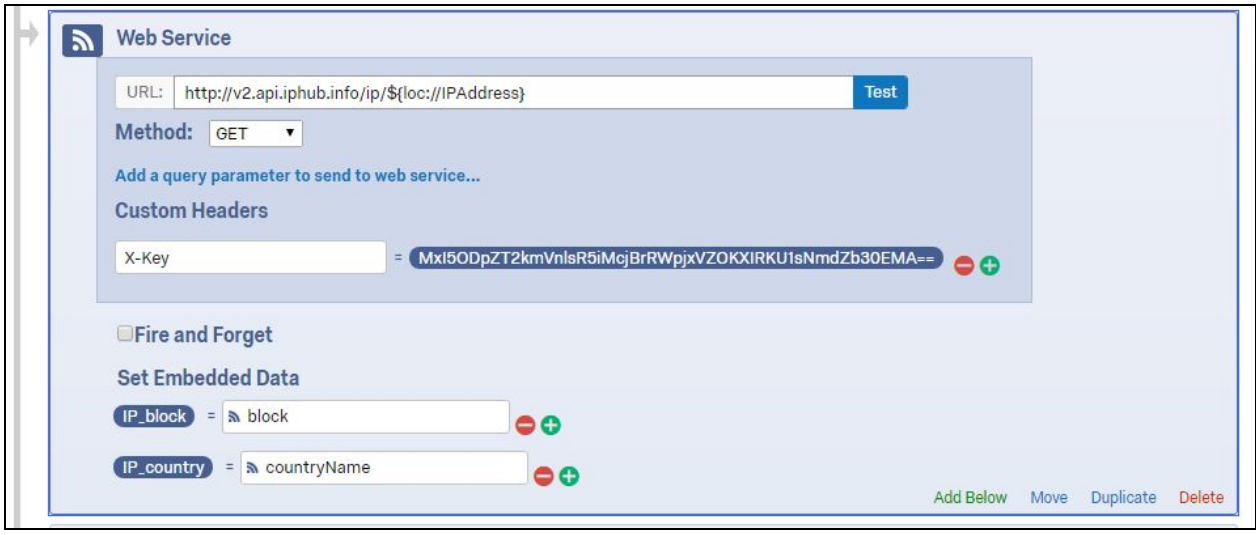
4. This is where we will make the call to IP Hub. In the “URL:” line, place in the following address: http://v2.api.iphub.info/ip/${loc://IPAddress}. The first part of this address calls the IP Hub API, the last part takes the IP address captured by Qualtrics and adds it to the API call. Make sure the “Method:” is set to “GET”.



5. Click on “Add a custom header to send to web service…” On the left-hand side, for “Header to Web Service…” type in X-Key. On the right-hand side, where it says “Set a Value Now”, type in your API key.



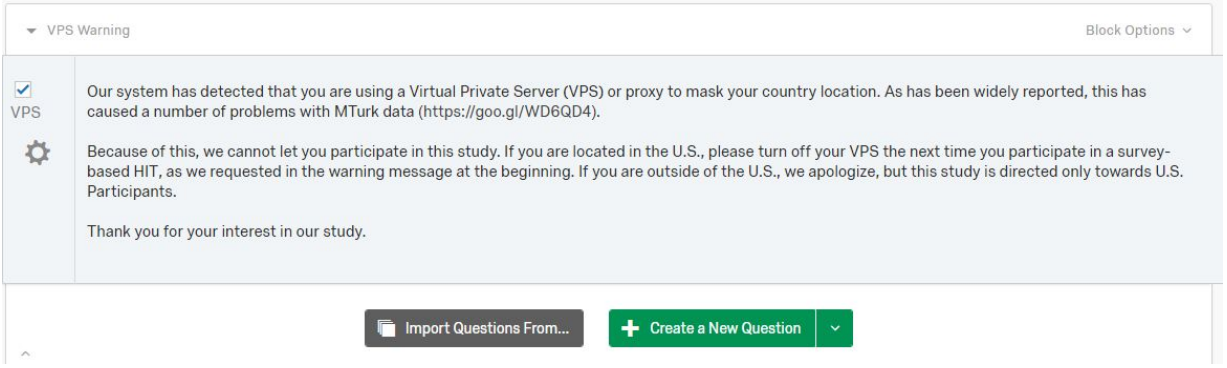
6. Next, click on “Add Embedded Data...” and add entries for IP_block and IP_country that correspond with the returned IP Hub fields block and countryName.



7. Click Save Flow. If you would like to screen out international or VPS actors, you can do this using the next few steps. Begin by setting up warnings explaining why they are not allowed to take the survey. These can be added as descriptive text questions in their own Block (or text entry questions if using the appeals procedure described at the end). This is both courteous and will prevent you from getting nasty emails. Below are the ones we used.

VPS Warning:

Qualtrics View



Survey View

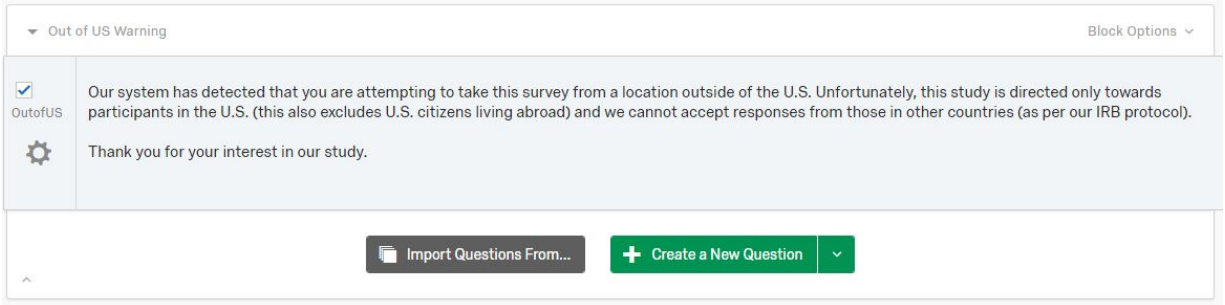
Our system has detected that you are using a Virtual Private Server (VPS) or proxy to mask your country location. As has been widely reported, this has caused a number of problems with MTurk data (https://goo.gl/WD6QD4).

Because of this, we cannot let you participate in this study. If you are located in the U.S., please turn off your VPS the next time you participate in a survey-based HIT, as we requested in the warning message at the beginning. If you are outside of the U.S., we apologize, but this study is directed only towards U.S. Participants.

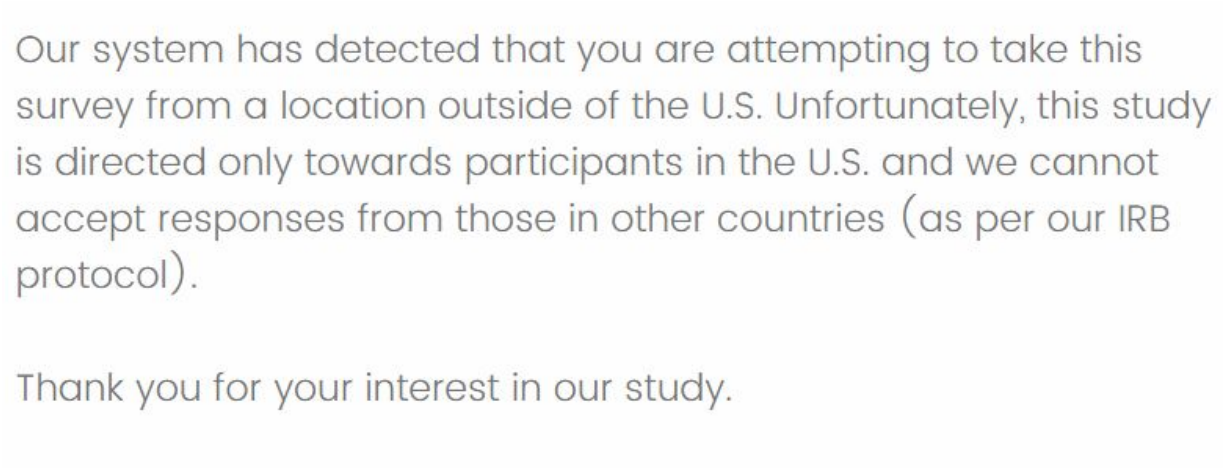
Thank you for your interest in our study.

Out of US Warning:

Qualtrics View

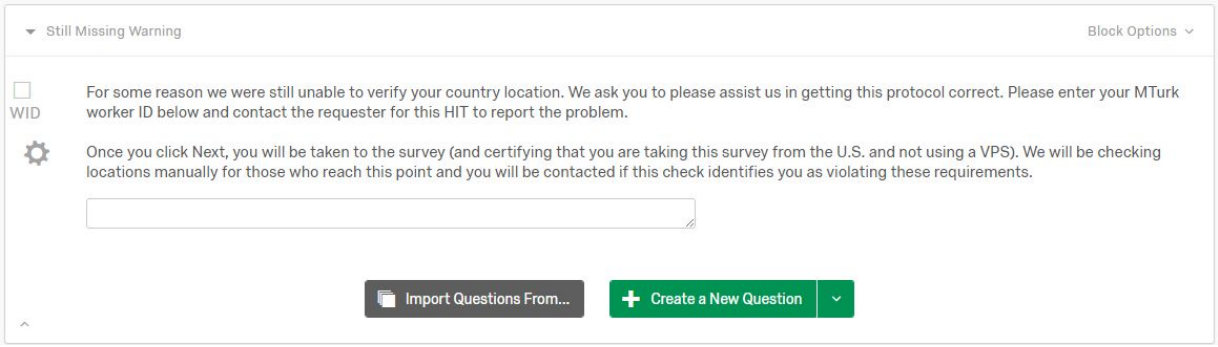


Survey View

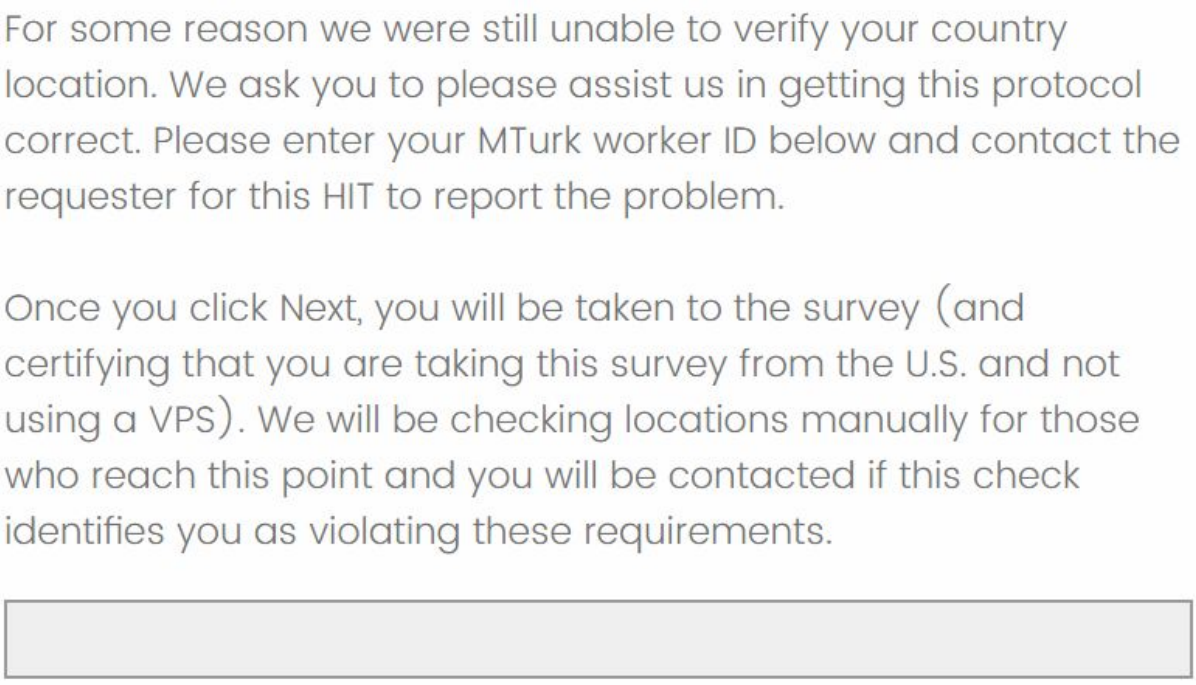


Still Missing Warning
(this message is added defensively. We find a small number of cases (about 1.6% in our pilot) the API lookup does not succeed and responses need to be checked after the survey is complete):

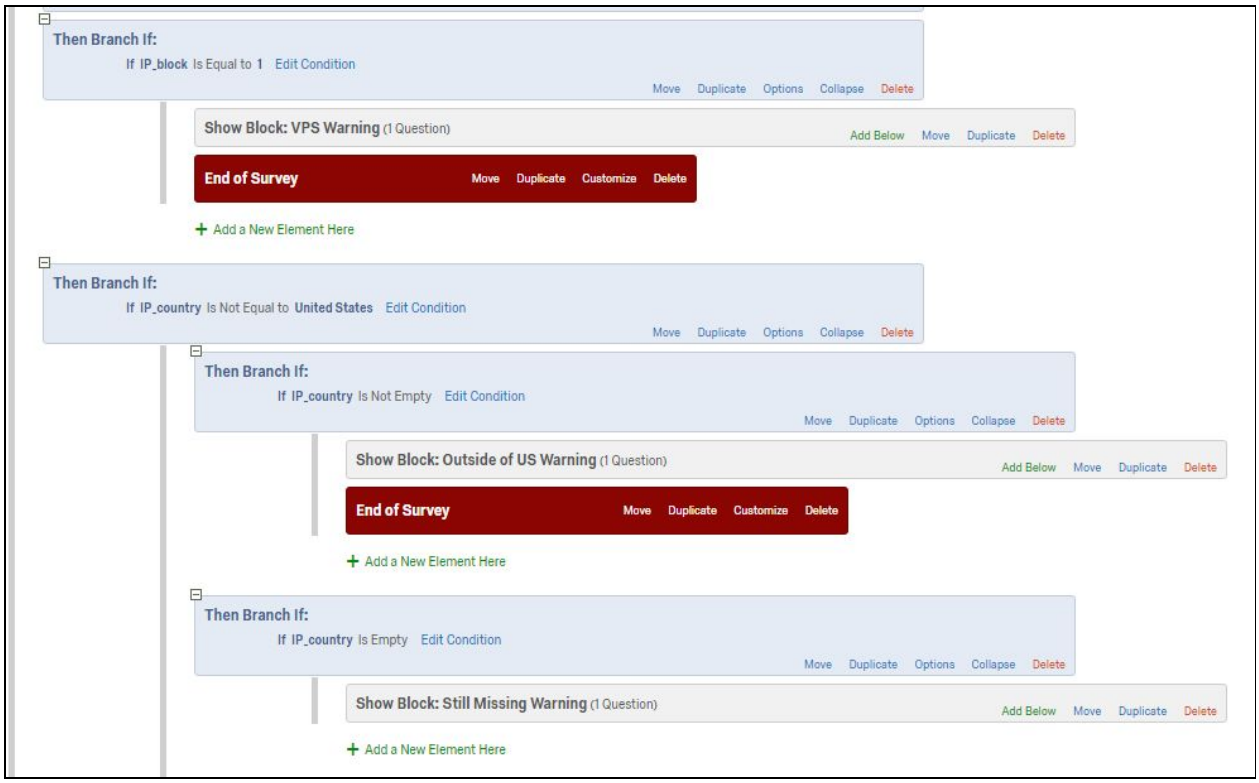
Qualtrics View



Survey View



8. Now go back to the Survey Flow. After the web service call we added earlier, add two Branches that respond to Embedded Data. For the first one, set it to activate “If IP_block is Equal to 1”. Move your VPS warning text underneath this branch and then add an End of Survey option below it. For the second Branch, set it to activate “If IP_country is Not Equal to United States”, then create two sub-Branches for “If IP_country is Not Empty” and “If IP_country is Empty”. Drag your out of US warning underneath under the first sub-Branch and add an End of Survey option below it. Under your second sub-Branch where IP_country is empty (this means that there was an error in the IP trace) drag your location missing warning. Now if anyone tries to access your survey from outside the US or from a server farm, they will be shown a warning and taken to the end of your survey. This part of your survey flow will look like the illustration below.



9. Save the new Survey Flow. Now nobody (or at least very few people) outside of the US or using a detected server farm should be able to take your survey.