An End-to-End Analysis of Covid-Themed Scams in the Wild

Behzad Ousat
Florida International University
Miami, USA
bousat@fiu.edu

Mohammad Ali Tofighi
Florida International University
Miami, USA
ali@cs.fiu.edu

Amin Kharraz
Florida International University
Miami, USA
ak@cs.fiu.edu

ABSTRACT

Covid19-themed attacks took the Internet by surprise in March 2020. Adversaries updated their attack strategies rapidly and started to exploit users' attention to this unprecedented event and distribute their malicious payloads. In this work, we perform a retrospective analysis of adversarial operations over the first four months from February 15th, 2020 to June 16th, 2020. By combining a variety of measurement perspectives, we perform a three-step analysis, by (1) analyzing the composition, growth, and reachability of Covid19-themed attack pages, (2) identifying the modus operandi of attackers, and (3) assessing the actual impact on end-users. Our measurements serve as a lens into the fragile parts of the Web ecosystem during a previously unseen attack. We argue that precipitous growth of Covid19-themed attacks in just a few weeks represents adversaries' technical and operational agility in adapting their attack strategies and also demonstrates how novice attack techniques can bypass common defense mechanisms and expose unsuspecting users to different forms of attacks. Drawing upon these analyses, we discuss what went poorly, in an effort to understand how the technical community can respond more effectively to such events in the future.

CCS CONCEPTS

• **Security and privacy → Social aspects of security and privacy**; **Web application security**.

KEYWORDS

Web Security, Social Engineering, Measurement

ACM Reference Format:
Behzad Ousat, Mohammad Ali Tofighi, and Amin Kharraz. 2023. An End-to-End Analysis of Covid-Themed Scams in the Wild. In *ACM ASIA Conference on Computer and Communications Security (ASIA CCS '23), July 10–14, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3579856.3582831

1 INTRODUCTION

For a long time, adversaries have been exploiting users' fear and attention to major news stories or regular national events (e.g., tax preparation season [3, 19]) to launch their attacks. In December 2019, a new opportunity arose due to CoronaVirus 2 (Covid19)– a severe acute respiratory syndrome [57]. As the unprecedented pandemic continued to grow, governments asked millions of residents to stay home to slow down the spread of the virus and flatten the curve of the pandemic [4]. Adversaries were keenly aware of the opportunities for abusing this situation to their advantage by targeting millions of remote employees who were not ready for this unprecedented phenomenon.

There have been several reports on the emergence of Covid19-themed scams during the pandemic and their consequential impacts on users and institutions. However, the details of these attacks, the modus operandi of attackers, and deployed tricks are mostly anecdotal. Most of these reports were relying mostly on ad-hoc procedures without providing sufficient insights about the type of malicious practices and level of sophistication of the reported threats [2, 5, 34, 36, 43].

Our work is guided by three primary research questions. First, how did users get exposed to such scam websites during their normal web browsing? Second, how prevalent were these forms of scams? And third, what was the potential impact of these scams on users? We begin our experiments by crawling the web in a targeted way. In particular, to provide an approximate view of the threat landscape, we used the Google Trends service [1] to build a set of common keywords relevant to Internet user searches at a global scale about Covid19. We incorporated Microsoft Cognitive Services [37] search API to crawl the list of websites associated with the trend list indexed by the search engine. We started the experiment before the lockdown in the US from February 15, 2020 to June 16, 2020 by visiting websites that were relevant to the collected Google trends terms. we also performed a subsequent analysis five months later in November 2020 to measure changes in the dynamics of the attack landscape. Armed with terabytes of web forensics data and web resources, we perform a retrospective analysis of Covid19-themed attacks by incorporating a diverse set of vantage points including JavaScript traces of attack pages, dynamic analysis reports of malicious binaries, document-based malware delivered on the scam page, and their corresponding filesystem and network traces.

Our analysis showed that Covid19-themed scam pages were easily accessible by normal users. Normal users could be exposed to Covid19-themed attacks by simply seeking information about the disease, such as how to buy related equipment or donate. We constructed more than 15K attack chains – the path that shows how a user is exposed to a Covid19-themed attack page. We observed that almost 50% of the attack pages were reachable from the Alexa top 60K websites. We also observed that Covid19-themed attacks became popular quite rapidly during the analysis time. For instance, the number of Covid19-themed attacks tripled in just eight days and increased tenfold in 30 days. We identified an abrupt increase in the number of malicious payloads as well as distinct attack cases after the nationwide stay-at-home advisory in the US on March

ASIA CCS '23, July 10–14, 2023, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0098-9/23/07…$15.00
https://doi.org/10.1145/3579856.3582831

15th. We identified 6,532 unique malicious binaries including 1,734 ransomware (26.5%) and 4,507 Remote Access Trojans (69%), other forms of suspicious payloads (4.5%). Sodinokibi, a ransomware family with self-propagating capabilities, and AgentTesla were among the most popular forms of attacks. The most common activities that we observed were filesystem exploration, keylogging attempts, and microphone access. These were the more dominant goals in campaigns we detected.

We observed that several scam campaigns started to emerge during this short period. We detected instances of Covid19-themed attacks from 10 campaigns that began less than 24 hours after the stay-at-home advisory. We identified 64 campaigns which ranged in size from 2 to 239 websites. We observed that almost all of the identified campaigns use a multitude of different sub-networks and hosting providers and hide their delivery servers behind CDNs, rendering blacklisting techniques less effective. Most of these campaigns used cheap TLDs, such as `.space`, `.club`, `.xyz`, and `.online` to generate different variations of addresses.

While the security problems identified by this work may not be representative of all the pages that users might visit on a global scale, they are indicators that abusing critical Internet services such as domain registries and cloud services can create a fertile environment for opportunistic adversaries. Perhaps our most important finding in this paper is practical evidence that reminds us once more that adversaries are agile and that they have an asymmetric power to quickly repurpose their tools once they identify a new opportunity. The results confirm the folk wisdom that Covid19-themed attacks emerged and continued to increase in type, reachability, and attack payload very rapidly [5, 34, 36, 43, 58]. However, our analysis also shows that the adversarial practices used in these attacks did not differ significantly compared to other forms of scams and social engineering attacks that we had seen before [29, 39] in terms of sophistication and evasive techniques. They neither deliver very different forms of malicious payloads.

Our hope is that this work serves to raise awareness about the importance of systematic approaches for repeated scanning and data cataloging to identify unsavory practices on a large scale for the web ecosystem. We also hope that our approach will prove useful to the web security community and opens the door to identifying emerging threats that go beyond what is routinely observed today. This paper makes the following contributions:

- A large scale measurement study of the end-to-end life cycle of COVID19-themed attacks. We performed a longitudinal data collection by generating 5 TBs of web forensics data including Javascript execution traces and network traffic.
- We identified 64 campaigns over four months consecutive crawling experiments. We compared Covid19-themed scams with phishing pages during the same scanning period to compare the dynamics of such scam trends in the wild.
- We collected a large dataset of malicious binaries relevant to COVID19. We identified 6,532 malicious binaries from 18 different malware families, including 669 previously unseen malware including Macro-based Documents, Trojans, and Potentially Unwanted Software(PUPs).

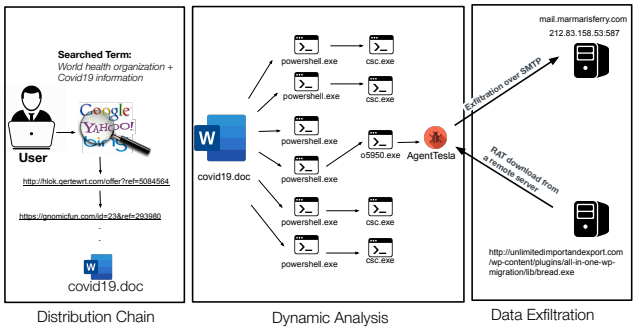

Figure 1: An example of Covid19-themed attack. Seeking guidance on how to apply for a loan during Covid19 directs the unsuspecting user to a malicious website that delivers AgentTesla – a remote access trojan.

2 BACKGROUND

The Covid19 pandemic and its global spread provided a unique opportunity for adversaries to abuse the panic and discomfort to launch large-scale attacks around the world. Reports of Covid19-themed attacks appeared as early as January 2020 [57], although it was not until early March 2020 when these attacks grabbed headlines with targeted attacks on biotechnology research firms or the US Health and Human Services website (hhs.gov). Several additional high-profile attacks later targeted end users and remote employees in Ukraine, China, Italy, and Spain. Massive social-engineering attacks on end users and government-sponsored attacks also made headlines in early April. Table 9 shows the timeline of some major incidents over the last three months. Throughout our study, we corroborate our findings with these reports and extent the public information surrounding Covid19-themed attacks.

Figure 1 illustrates a multi-step process to analyze Covid19-themed attacks at scale. In step one of this example, a user uses a search engine to find some guidance about Covid19. By clicking on one of the search results, after 8 page redirections, he is redirected to a landing page (x.com) and is encouraged to open the document. The document contains macros that launch several PowerShell processes. One of these processes connects to a remote server and downloads a binary that is executed by launching this command. Note that the example we discussed here is not synthetic and is a simplified version of a real-world attack. Our experiments, discussed at length in Section 4, show that these attacks occur quite frequently. To perform a systematic analysis that contains all the required information, our analysis contains three independent components. The web scanner in the first step allows us to record the interaction of websites using an instrumented browser and reconstruct the attack distribution chain. In the second step, we analyze the payload and catalog malicious payloads. In the last step, we incorporate our knowledge and answer questions about the actual attack and underlying ecosystem of Covid19-themed attacks.

3 METHODOLOGY

In this section, we describe our approach on how to systematically measure Covid19-themed attacks. Figure 2 summarizes the pipeline

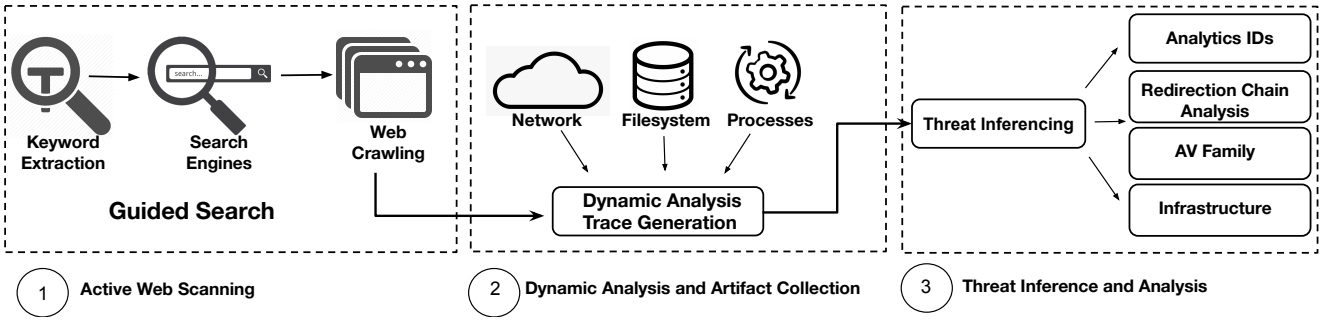

Figure 2: A summarized version of our experiment pipeline.

of our experiment. Our study leverages a variety of vantage points, including an active web scanning platform, JavaScript execution traces, filesystem, and network logs of collected binaries.

3.1 Guided Search

Our guided search consists of two main components: (1) a systematic way to simulate how users browse the web and create a list of websites to visit, and (2) a crawling module to perform repeated scanning and generate fine-grained artifacts.

3.1.1 ***Keyword Extraction.*** We took advantage of the findings of recent studies on web-based social engineering attacks [29, 30, 40, 55], and tried to incorporate websites that used a combination of *deception* and *persuasion* to attract users. We specifically looked for pages that try to attract normal users by embedding enticing content and encouraging users to make risky decisions (e.g., downloading a file). To construct a dataset of real-world examples, we incorporated *Google Trends* service to build trendy keywords information about Covid19. Since trend keywords are constructed based on real users' search behavior and are indexed based on their popularity, they can be viewed as a proxy of more prevalent real-world searched items in the wild. After collecting a list of popular keywords relevant to Covid19 across different categories, we used *Microsoft Cognitive Services* to automatically make queries using an assigned web search API [37]. In Section 4.1.1, we explain how we generated a list of trend words and the corresponding search engine results.

3.1.2 ***Instrumentation and Trace Collection.*** We employed a customized instrumented browser to monitor the interaction of web content with browser resources and reconstruct the payload delivery paths. such as resource usage and network communication. We leverage the Chromium Debugging Protocol to access nearly all the functionality of DevTools [18] as well as DOM and DOM Events of a page for data collection. More specifically, the crawler collects browser request and response traces, JavaScript execution logs, and the source code of both inline and dynamically loaded JavaScript code. This set of information was sufficient to locate the functions being executed, external code referenced by the function, and the redirection chains – the path that shows how a user is exposed to a Covid19-themed attack page.

3.1.3 ***Active Web Scanning.*** To run the experiments at scale, we deployed the instrumented scanner across 40 virtual machines for distributed scanning. To increase the level of interaction, the scanner scrolls downwards to trigger event listeners that are responsible for the loading of dynamic content. The reason for running interactive sessions was that modern websites usually set several event listeners which are triggered based on the level of interaction with the website. This approach would make site visits more similar to realistic scenarios and can potentially increase the chance of triggering specific events that lead to page redirection. The crawler remains on each page for a fixed period of 30 seconds before clearing its entire state, restarting, and then proceeding to the next site. We empirically found that 30 seconds was sufficient time for websites to retrieve necessary page resources, compile and run javascripts and construct the page. Our analysis of 1,000 randomly selected websites shows that the run-time overhead of the instrumentation layer is approximately 0.5%.

3.2 Analyzing Delivered Contents

One question that arises is that how we can determine the type of malicious practices on the landing page. The adversarial attempts can occur in different forms, such as delivering a malicious payload or exposing web users to different forms of scams or social engineering web pages. In this section, we explain how we determine the type of malicious practice in a delivery chain.

3.2.1 ***Cataloging Delivered Scams.*** Conducting a manual process to verify landing pages and cataloging their types is indeed a non-trivial task. To automatically identify the type of a given page, we leverage structural similarity testing [59] to cluster pages based on their visual appearance. We used structural similarity since our initial analysis showed that Structural similarity works well to identify two images with minor changes in scale, ratio alteration, or color. We started by manually clustering 1,000 different unique scam webpages into four types of Covid19-themed scams: (1) Covid19 donation scams, (2) emergency loan scams, (3) employment opportunities due to lockdown, and (4) Covid19-themed pharmaceutical scams. In all the observed cases, potential victims had to fill out a form and share PII information including Social Security Number, name, physical, and email addresses. Empirically, we assigned the threshold value t= 0.86 since at this value we were able to generate tighter clusters of websites representing different classes of threats. We achieved a precision of 99.2% and recall of 98.2% compared to the manually generated ground truth of 1,000

screenshots. The procedure to perform manual labeling took 18 hours of work. However, it saved us hundreds of hours of manual work for the test on all potential cases.

3.2.2 ***Cataloging Delivered Payloads.*** The second phase of our analysis pipeline seeks to answer what payloads adversaries intend to deliver in Covid19-themed attacks. We partnered with a well-known anti-malware company that offers bare-metal sandboxing environment to perform dynamic analysis on the collected malicious code. This was a significant improvement in our analysis because, unlike virtual analysis environments, bare-metal sandboxes do not leave much artifacts during analysis time. This makes most of the anti-reverse engineering techniques (e.g., checking the debugging status, virtualized hardware, and drivers) ineffective. Dynamic analysis reports contained fine-grained forensics analysis data about all filesystem and network activities as well as actual dropped binary payloads, spawned processes and threads, and executed commands. The collected artifacts also provided visibility over the resource usage such as by installing low-level hooks for microphone, camera, keyboard APIs. We realized that this piece of information on run-time behavior was useful to identify remote access trojans and backdoors, which will be discussed in Section 4.

Category Selection A question that arises after collecting the artifacts was to answer what would be the core behavior of a given artifact. At first glance, community vetting mechanisms such as VirusTotal [49] would appear to be an ideal case. We found this approach not very affective because the generated reports about the submitted samples were often very abstract and did not provide sufficient information about the actual run-time behavior of a malicious binary. We empirically tested the community vetting approach on 250 collected samples. 187 (75%) of the submitted samples were labeled as generic malware by in VirusTotal confirming our intuition. An alternative approach was to categorize the samples based on the run-time behavior. Starting with the raw trace data from our data collection, we considered 32 related forensic events about the filesystem, network, and processes that seemed to be useful in attributing the run-time behavior of a given binary. However, we narrowed down our list to 15 forensic events as we could not prove that all those features were distinctive for malicious payloads. In the following, we briefly explain these events and describe why we used them.

Executing Payloads through APIs: As we mentioned earlier, the initial payload in modern malware attacks usually does not contain the actual malicious payload. Thus, after a successful execution, the malicious process would programmatically download and execute additional payloads using specific Windows APIs. Functions such as Windows API CreateProcess and LibraryLoad will allow a process to start other processes with proper path and arguments. We search for a list of Windows API calls by checking dynamic analysis reports and use the number of API calls. The list is shown in Table 10.

Interacting with Command-line Tools: Adversaries can use command-line interfaces to interact with the target system during the course of an operation. For instance, InstallUtil is a command-line utility that allows installing and removing resources by incorporating installer components in .NET binaries. InstallUtil is a digitally signed utility, and adversaries may use it as a proxy to execute code and bypass specific policy-based approaches such as process whitelisting.

Loading Scripts: The malicious process may use VBScript or PowerShell scripts to run tasks automatically. We search for the execution of VBscript macros as well as PowerShell scripts by looking for Windows script (WScript) which provides an environment to execute scripts in a variety of languages.

Fetching System Information: An important step before delivering the actual malicious payload is to fingerprint the environment and query detailed information about the operating system, hardware architecture, software patches, version, hotfixes, and service packs. Example commands to obtain this sort of information include *ver, Systeminfo, and dir.* Another form of system information gathering in Windows malware binaries is to interact with the Windows Registry looking for specific software names (e.g., virtualbox, vmware) and configuration. That being said, fetching system information alone is not necessarily an indication of malicious behavior, but our empirical analysis showed that considering this feature along with other features can improve the detection capability.

Manifesting Persistent Behavior: Malicious payloads can go one step further and modify the Windows Registry information or other parts of the operating system to make permanent changes. For instance, adding an entry to the *run keys* in the Registry or startup folder will cause the program to be executed after a successful login. These programs will be executed under the context of the current user and will have the associated permissions level of the account.

Dropping Remote Files: As we showed in Section 2, the initial binary can drop binaries after being executed. The dropped files can be uncompiled code, encoded, or embedded within other files. These files may also be delivered in formats unrecognizable and inherently benign to the native OS. For instance, we observed 170 samples that dropped 425 files in 27 different file extensions. The dropped files are then compiled by incorporating native utilities such as csc.exe, the Microsoft's C# language command-line compiler for the .NET Framework, or GCC/MinGW. We observed several samples from Sodinokibi and Emotet that incorporate this technique to compile and start the actual malicious operation.

Leveraging Known Ports: While there are specific malware cases that use uncommon ports for communication, scanning, or delivering malicious payloads, a large number of malware binaries communicate over a commonly used port to blend with normal network traffic and bypass firewalls or anomaly detection systems. HTTP, HTTPS, and SMTP are among the most commonly used ports for communication and data exfiltration.

DNS Queries: In line with the previous features, we also take into account the number of DNS queries. Malicious binaries make DNS queries for several reasons. The malicious process might use a DGA engine to bypass blacklists or might contain a scanning engine to find other targets for infections – a behavior similar to Wannacry attack for self-replication.

3.2.3 ***Artifact Clustering.*** We created a pool of labeled dataset that contains the corresponding traces of 300 malicious binaries shown in table 11. We created a vector of features defined above and applied the Locality Sensitive Hashing (LSH) algorithm [23] to these vectors to find similar artifacts. This resulted in 10 clusters to start the artifact cataloging. The approach achieved an accuracy

of 95.7% at 0.6% false positive. We performed an experiment on labeled malicious dataset to measure the relative contribution of the features used in process. We used a recursive feature elimination (RFE) approach to determine the significance of each feature. We divided the feature set into three different categories: Process-based, Network-based, and Filesystem-based features. As a first step, we incorporated all the feature categories while measuring the FP and TP rates. Then, at each step, a feature with the minimum weight was removed and the FP and TP rates were calculated to quantify the contribution of each feature. Table 1 ranks all the features with the most important one at the top. The capitalized letters in the second column indicate the feature categories: P for Process-based features, N for Network-based features, and F for Filesystem-based features. We calculate the score ratio by dividing the score values by the largest one. The score ratio simply shows the contribution of each feature in identifying positive and negative cases. The results of the experiment shows that 4 out of 6 Process-based features are among the top ten features.

Table 1: Ranking of feature importance (P for Process-based, N for Network-based, and F for Filesystem-based category).

Rank	Cat	Feature Name	Score Ratio
1	P	Payload Execution through APIs	100%
2	P	Number of loaded scripts	86.2%
3	P	Interacting with Command-line Tools	70.4%
4	P	Write operation in registry	63.4%
5	F	Mean size of dropped files	30.5%
6	F	Number of file extensions	28.2%
7	N	Number of DNS Queries	27.6%
8	F	Number of dropped files	22.5%
9	N	Payload size sent mean	11.2%
10	N	Payload size sent max	9.6%
11	P	Fetching System Information	8.5%
12	P	Fetching Registry Information	8.3%
13	N	Payload size received mean	7.2%
14	N	Payload size received max	5.3%
15	N	Leveraging Known Ports	3.2%

4 UNDERSTANDING COVID19-THEMED ATTACKS AT SCALE

In this section, we apply the metrics we described in Section 3.2.2 on dynamic analysis traces and we first pre-filter the less relevant payloads or those that did not show any activity in the scope of this work. We then answer questions about Covid19-themed attack strategies as well as the underlying ecosystem and the distribution chain, and campaigns involved in this emerging threat.

4.1 Dataset

4.1.1 ***Covid19 Pages.*** We created a list of 1,000 search items that were queried by real users at least 300,000 times in six main search categories such as business, technology, and news, then extracted the first 10 pages of search results of each query. We used the first 5 pages since as we increased the number of search result pages, the results became less relevant. Table 2 shows the distribution of the keywords across the more relevant categories. Microsoft Cognitive service provided 10 results per page. We used these initial indexed pages as the initial seed for our scanning experiments. We observed that 5,272 (4.4%) of the pages were not reachable during the experiment. Consequently, the seed page contained 114,728 pages. As an example, the search term *"N95 masks shipping"* that was indexed as a popular searched word, led us to `https://b2j-system-err.xyz/` where it required user information to send the package. Several cases asked users to download and install desktop or mobile applications from third-party app stores. Figure 3 shows an example that the page simply takes the IP address of the visitor and encourages him to install an application to get accurate updates about the pandemic in his geographical location, but it was a dropper that delivered Sodinokibi ransomware. In another example, users are encouraged to download a mobile application that could determine the body temperature or blood oxygen level, but VirusTotal reported it as a mobile keylogger. The web scanner crawled these websites by clicking on the links, recording the redirection chains, and taking a screenshot of the landing page.

Table 2: The distribution of Google Trend keywords across the more relevant categories.

Category	Google Trend Keywords	URLs(#)
Business	100	18,780
Health	212	19,940
News	187	18,898
People& Society	119	18,829
Science	230	19,959
Technology	152	18,322
Total	1,000	114,728

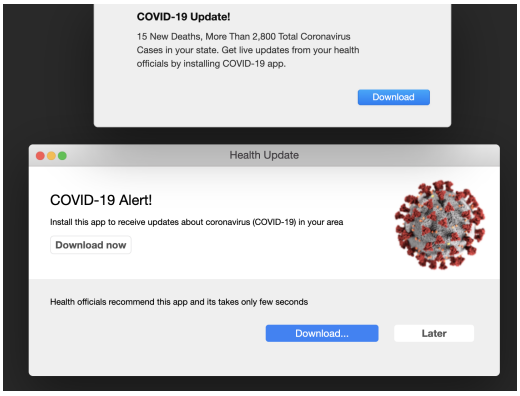

Figure 3: An Example of Covid19-themed attack that installs Sodinokibi ransomware after downloading and installing the payload.

4.1.2 ***Crawling Experiments.*** We performed a periodic web scanning experiment for four months from February 15, 2020 to June 16, 2020 every four days. For each URL, the scanner randomly

selected three links and clicked on the link to simulate user interaction/navigation on the page. The scanner visited 15,205,042 URLs during the four-month experiment. We collected 53,342 payloads including compressed and archive files, executables, extensions, and documents (e.g., PDF, Word, Excel). The summary of the collected files is provided in Table 3.

Table 3: Summary of the large-scale experiment over a four-month experiment.

Exp. Summary for Covid19 Pages	(#)
Crawling Period	Feb 15th – June 16th
Initial Seed using Google Trends	114,728
Crawled pages	15,205,042
Analyzed Payloads	53,342
Malicious Payloads	6,532
Covid19-based Scam Pages	9,993
Identified Campaigns	64
Websites in largest Campaign	239

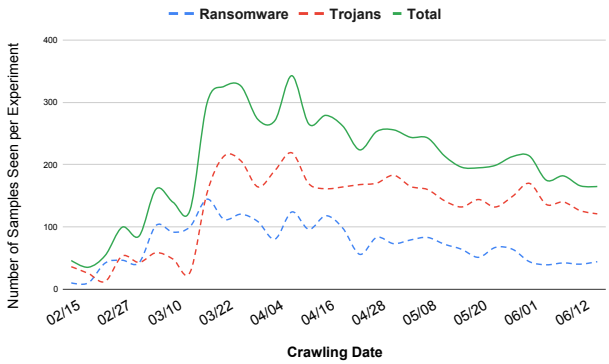

Figure 4: Number of detected ransomware and trojan malwares during the experiments

Our analysis shows that there was a significant increase in the number of ransomware as well as Remote Access Trojans (RATs) across the 6,532 malicious payloads. We observed more than tenfold growth over one month from 47 ransomware and Remote Access Trojans (RATs) in our first crawl on February 15th to 493 cases on the eighth crawl on March 14th. The results of our analysis confirm the folk wisdom that delivering malware attacks, as well as scam pages, have an abrupt increase immediately *after* the nationwide stay-home advisory in the US starting from March 15th. In particular, we noticed a sudden increase in the number of detected cases between two consequent crawling experiments on March 14th and March 18th as shown in Figure 6 in the Appendix Section. That is, we started to see the emergence of NanoCore and njRAT trojans during these time periods. Furthermore, our average malware collection per experiment was 64 samples before March 14, 2020. This number increased to an average of 340 samples for the next crawling experiments. Figure 4, in the Appendix section, shows the number of ransomware and trojan malwares detected in each crawl cycle of the experiment. Our analysis shows an increasing trend regarding the mentioned malware categories used in Covid19-themed attacks. Table 4 presents the distribution of different types of malware according to all experiment cycles.

4.2 Potential Impacts on Users

We observed that exposing users to scam pages as well as delivering ransomware and remote access trojans are the most common types of malicious practices in Covid19-themed attacks. In the following, we discuss the attacks, the underlying ecosystem, and the campaigns involved in those operations.

4.2.1 ***Social Engineering and Deceptive Advertisements.*** As mentioned earlier, adversaries could use a variety of techniques to monetize their operations (e.g., pop-ups, injected ads, traffic redirection). We observed in 56% of the page redirections, the user is exposed to several overlay ads and widgets which were transparently injected into a page. In this paper, our goal is to increase the level of interaction with these elements since these websites often present security threats including scams and identity fraud. To this end, we extended the scanner module to search for iframe elements (i.e., overlay ads) of the visiting pages. We empirically knew that this process can include several layers of page redirection due to the nature of JavaScript and adversaries' desire to stay undetected. As we mentioned in Section 3, the active web scanner records almost all visited URLs and their corresponding HTML contents during the redirection process. It also automatically captures screenshots of every opened webpage. At the end of the process, we collected the screenshots of 109,384 pages that were opened because of clicking on advertisements. Capturing a snapshot of the fetched pages assists us to approximate what a user would have been exposed to while interacting with the ads. We observed that 9,993 of these snapshots contained specific Covid19-based attacks. These attacks mainly revolved around selling fake Covid19 treatment kits, providing monetary assistance such as loan or credit deferral, and recruitment opportunities after providing Personally Identifiable Information (PII) information.

4.2.2 ***Malware Payloads.*** Clicking on overlay ads and widgets resulted in fetching 53,342 unique files. The cataloging mechanism identified 6,532 (12.2%) of the cases as malicious. 4,507 (69%) of the detected cases were identified as Remote Access Trojans. AgentTesla, Emotet, and Lokibot constituted 44% of all RAT samples. We also observed that Emotet samples in 11% of cases were manifesting ransomware-like behavior by encrypting user data. In addition to RATs, we also observed common types of ransomware families. Sodinokibi and Ryuk were seen in 1,734 (26.5%) of the cases. When submitting the detected cases to VirusTotal to receive community labeling, we observed that 10.25% of the samples were submitted for the first time to VirusTotal. Although this experiment can potentially show the effectiveness of our approach in collecting previously unseen samples, we believe that this could happen for multiple reasons (e.g., AV scanners may want to maintain a competitive advantage by not sharing their entire samples catalog). That being said, we do not have enough data, nor have investigated enough about the coverage of community vetting scanners.

Table 5 shows detailed information about most observed activities in the collected samples. The activities we observed were filesystem exploration, microphone access, and keylogging attempts. We observed attempts to access decoy passwords in 78% of identified samples. The victim filesystem was explored in 70% of all the cases. The prevalence of actions that attempt to collect information about the user as well as their files suggests that, unsurprisingly, surveillance is a dominant goal of Covid19-themed attacks.

Table 4: The distribution of malicious payload in Covid19-themed attacks. AgentTesla and Emotet were the more common types of RATs in our experiment. 404 (9.8%) of the collected samples were submitted to VirusTotal for the first time.

Malware	Occurrences	New to VT?
Ransomware	**1,734 (26.5%)**	**189 (10.9%)**
Dharma	353 (5.4%)	47 (13.3%)
Ryuk	451 (6.9%)	19 (4.2%)
Sodinokibi	557 (8.5%)	49 (8.8%)
Ursnif	373 (5.7%)	74 (19.8%)
RAT	**4,507(69%)**	**480 (10.7%)**
AgentTesla	1,315 (20.1%)	136 (13.3%)
Azorult	234 (3.6%)	4 (1.7%)
DarkComet	77 (1.2%)	2 (2.6%)
Emotet	1,047 (16%)	157 (15%)
Formbook	132 (2%)	3 (2.2%)
Qausar RAT	59 (0.9%)	0 (0.0%)
Lokibot	592 (9%)	52 (8.8%)
NanoCore	361 (5.5%)	33 (9.1%)
njRAT	346 (5.3%)	41 (11.8%)
Remcos	89 (1.3%)	3 (3.4%)
Trickbot	260 (4%)	40 (15.4%)
Browsing Hijackers	**291 (4.5%)**	**7 (2.4%)**
Musix search	121 (1.9%)	2 (1.6%)
Search by zoom	88 (1.3%)	3 (3.4%)
Secure Search	75 (1.1%)	1 (1.3%)
iMove	7(1%)	(0%)
Total	**6,532**	**669**

4.2.3 ***Redirection Analysis.*** Beyond analyzing social engineering attacks and malware delivery, we investigate how users are exposed to the attack pages during their normal browsing. We performed a reachability analysis that looks into the redirection chain of each attack incident from an initial search page to the actual attack page. Each redirection chain, as we mentioned earlier in Section 3, contains a sequence of URLs that shows how the web scanner reached the attack page. Table 12 shows a list of the top third-party entities that redirected users to the scam and malware pages. Some of these third-party code providers such as *adcash.com* and *srickyads.com* have been abused by malicious advertisements in the past [27, 40]. *Doubleclick.net* was also observed in 11% of the redirection chains. This is likely since Covid19-themed attack pages are a new practice and look less suspicious compared to other social engineering attacks for which more mature techniques are being used to detect them. We extended our analysis by checking

Table 5: Key logging, filesystem exploration, and audio capturing attempts were very popular among payloads delivered in Covid19-themed attacks.

Category	#	%	Major Families
Audio Capturing			
Record Microphone	3,135	(48%)	AgentTesla, DarkComet
Camera Capturing			
Camera access	2,155	(33%)	njRAT, DarkComet
Evasive Execution			
Execution thr. APIs	6,010	(92%)	All families
PowerShell	4,442	(68%)	All families
Compiling new files	3,593	(55%)	All families
Network Activity			
IP/Port Scanning	1,829	(28%)	Emotet, Sodinokibi
File fetching	6,336	(97%)	All families
SMTP traffic	1,045	(16%)	AgentTesla, DarkComet, njRat
Uncommon port	209	(3.2%)	Qausar RAT, Emotet
Filesystem Activity			
Download File	5,813	(89%)	All families
Encrypt File	1,698	(26%)	Ransomware families, Emotet
Search for File	5,030	(77%)	All Ransomware, Emotet, AgentTesla
Spyware Activity			
Stored password	5,291	(81%)	AgentTesla, Emotet, njRat, DarkComet
Keylogger	4,050	(62%)	Emotet, njRat, DarkComet

the reputation of the first node of each redirection chain. The analysis shows that more than 50% of the observed cases were reachable from the Alexa top 60K websites. Figure 7 illustrates the summary of the reachability analysis.

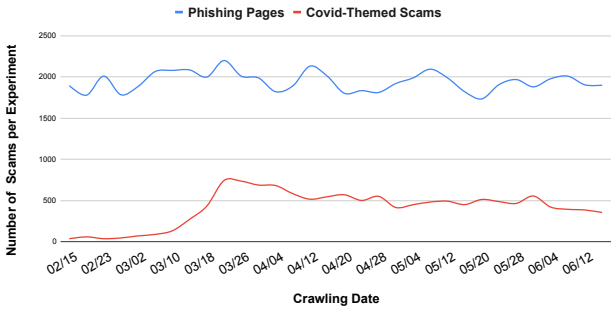

Figure 5: The longitudinal data on the dynamics of Covid19-themed attacks vs common phishing websites. Unlike Covid19-themed threat, the phishing website trend does not seem to change significantly.

4.3 Comparison with Phishing Websites

We observed that the number of Covid19-themed attack pages had increased abruptly and stayed almost constant during the analysis time frame. However, one question that arises is whether the observed trend has been seen only in Covid19-themed attacks or other social engineering attacks also manifested a similar trend. We ran an experiment on phishing websites during the same analysis window to evaluate how Covid19-themed attacks evolved compared to a more known social engineering attack. We used phishing websites as they are often considered as a classic social

engineering practice and can be used as a reference point when analyzing dynamic changes such as growth rate, underlying infrastructure, and daily samples. To this end, we used PhishTank daily list [6], a public seed for phishing websites. Over the course of the experiment, we performed daily queries to collect a list of reported phishing pages. We collected 62,144 phishing pages, on average 517 URLs per day, from February 15, 2020 to June 16, 2020. We used the list to connect to phishing websites and collect the corresponding details such as the hosting infrastructure and executed code in the browser. We removed 17% of pages from the list because they were not reachable during our scanning attempts. We also removed 6.5% of pages, which were very relevant to Covid-themed attacks, using the technique discussed in Section 3.2.1.

Figure 5 summarizes the result of the experiment. Our analysis of the PhishTank dataset does not suggest a significant change in the number of daily submitted phishing pages during the analysis timeline. Although we did observe small changes in the number of URLs submitted per day, the numbers did not diverge too much during the analysis time (i.e., 517 URLs on average per day). This trend was different for Covid19-themed websites at least at the beginning of the analysis. As shown, the number of incidents, although still smaller than phishing websites, started to grow in early March 2020 and peaked at March 26, 2020 and stayed almost constant until the end of the experiment timeline. The main takeaway from this experiment is that we did not observe a significant change in the number of phishing websites reported by the common phishing list provider in the first three months of lockdown. However, a new form of scam began to emerge quickly and remained active during the course of the experiment. We do not have enough evidence nor did we investigated other forms scams to make a generalizable claim on other trends that might have been occurred during this time period.

4.4 Experiments in Different Time Windows

Our longitudinal data does suggest that Covid19-themed attacks started to increase significantly right after in-home advisory notification in the US and continued to stay constant over the course of the experiment. We did another experiment to verify if this observation stayed consistent over a larger time span. To this end, we ran one more experimental snapshot for 35 days, five months after the last crawling experiment from November 15 2020 to December 20 2020 using the same trend keywords discussed in Table 2. Although the crawling time window is less than 30% of the reference experiment time, the number of malicious payloads collected following the same pipeline decreased from 12.2% to 3.6% (36 out of every 1,000 collected samples) as shown in Table 6. The trend is similar in the number of scam pages. There might be several reasons for this issue. For instance, keyword trends might have changed significantly during the five months prior to the second experiment or the attack strategy among adversaries might have been updated, and new trends might have emerged. We do not have sufficient data to elaborate in great detail on when this trend started to diminish or what were the underlying reasons for this decrease.

Table 6: Scanning the web for Covid19-themed attacks over two time window. Although the crawling window is not the same, we noticed significant decrease in the number of observed threats using the same trend keywords.

Scanning Details	**Exp.#1**	**Exp.#2**
Crawling Window	122 Days	35 Days
# of Analyzed Payloads	53,342	3,982
# of Malicious Payloads	6,532 (12.2%)	143 (3.6%)
# of Analyzed Pages	15,205,042	3,089,944
# of Covid19-based Scam Pages	9,993 (0.07%)	345 (0.01%)

5 OPERATIONAL INSIGHTS

As a second case study, we dig into the operations of Covid19-themed attacks to analysis the underlying infrastructure, identify campaigns that were involved in delivering malicious payloads as well as their underlying infrastructure.

5.0.1 ***Underlying Infrastructure.*** We also perform an analysis of the underlying ecosystem of Covid19-themed attacks, with a focus on gaining insight into perpetrators' delivery infrastructure. In particular, we would like to answer what infrastructure is used to conduct the operations and who operates it. To this end, we first extracted all the corresponding servers in the delivery chains that resulted in malware downloads or scam pages. This resulted in 6,243 unique domains. These attack delivery domains resolved to 862 unique IP addresses. We then mapped the IP addresses to AS names and found that 32% of the delivery servers were using Godaddy to host their servers. We also found 932 /24 corresponding network addresses where 80% of the remote servers resolved to 34 /24 network addresses.

We also analyzed the distribution of countries in which the attack delivery servers were hosted. Table 13 shows the geographical distribution by country. Overall, the US was the most popular hosting location for Covid19-themed attack delivery servers, accounting for approximately 60% of all the delivery servers we observed.

Our results suggest that the Covid19-themed attack ecosystem often uses shared infrastructure and network addresses to carry out its attacks. Similar to more traditional malware, having a large enough corpus of related domains can link seemingly unrelated malware instances and shed light on malware operators. Table 8 summarizes the characteristics of the six largest campaigns (grouped by effective second-level domain). As shown, each campaign uses a multitude of different subnetworks and hosting providers. In addition to using relatively diverse infrastructures, these distribution campaigns often hide their delivery servers behind CDNs, rendering blacklisting techniques less effective. Most of these campaigns use cheap TLDs, such as `.space`, `.club`, `.xyz`, and `.online` to generate different variations of addresses.

5.1 Campaign Analysis

We illustrated how malicious payloads are distributed and accessed by analyzing the redirection chains. In this section, we seek to identify the main entities that distribute Covid19-themed attacks. To this end, we focus primarily on the last page of the redirection chain, as they are responsible for delivering all types of malicious

payload we discussed in this paper to end-users. To group pages that are controlled by the same entities, we make use of web analytics tools. Adversaries similar to website publishers may take advantage of third-party analytics platforms such as Google Analytics, Yandex, and ClickTale to track, analyze, and report traffic volume. Therefore, if we find an identifier number from any of the third-party analytics on the malware delivery pages, it is highly likely that the analytics ID would be used in other delivery pages if they are controlled by the same entity. Third-party analytics services can then use the analytics ID and infer that websites with the same analytics ID are related to some extent even if the domain names do not share any substring, are hosted on different servers, and conceal registration information using privacy solutions. Note that this approach would not be quite effective if adversaries used a new analytics ID for each website under their control. We identify a total of 1,031 unique analytics IDs associated with delivery pages. We were able to cluster 492 domains (2,389 URLs) into 64 clusters that ranged from 2 to 239 pages. Table 7 illustrates a subset of the more prevalent third-party analytics services on distribution websites.

Table 7: The most common analytics services in distribution pages. We identified 1,031 unique IDs and 64 clusters ranging from 2 to 239 domains that were used to deliver malware and scam distribution pages.

Analytics	# of Analytics IDs	# of URLs with IDs
Clicky	45	110
Google Analytics	289	1,322
Google Tag Manager	138	217
Insp	34	93
HotJar	131	119
Qauntcast	43	58
MouseFlow	29	23
StatCounter	39	112
Site Catalyst	53	52
Yandex	82	139
Others	148	144
Total	1,031	2,389

To get a flavor for the behavior of a Covid19-themed campaign, we took a closer look at the largest campaign with 234 domains. all the clustered domains were registered under anonymous WHOIS services and pointed to a set of 44 IP addresses found in 23 ASes, which were mostly cloud-hosting services. The IPs were globally distributed in 4 countries including the US, Germany, China, and Canada. All IPs point to HTTP webservers running Openresty, Nginx, or Apached. The campaign's usage of multiple anonymous WHOIS services, globally distributed cloud hosting, and a large number of fungible, human-meaningless domains hint at evasion techniques characteristic of other web-based malicious practices, such as malware or PUPs.

Table 8 shows a set of more prevalent campaigns. The table also provides details including the first seen date in our periodic crawling and the main type of threat they introduced. We observed that 40 (80%) of the identified campaigns were seen for the first time after one month of the experiment between 02/10/2020 and 02/22/2020. Although we do not have enough evidence to scientifically measure the correlation between the time of the first stay-at-home advisory on March 15th, we noticed a 160% increase in the number of campaigns involved between 02/10/2020 and 02/22/2020.

6 DISCUSSION

While the global outreach of Covid19-themed attacks qualifies them as a contemporary web threat, it is not immediately clear what would be the correlation between Covid19-themed attacks and other types of web threats, what can be learned from this threat and its widespread impact, and how the security community should react in the future to new emerging threats of this sort. In this section, we discuss the implications of our investigation and make recommendations for potential routes forward.

6.1 Community Consensus

Defining rigorous web auditing mechanisms requires community consensus. We posit that the community needs a consensus on where and how auditing mechanisms should take place at the web-scale. The web ecosystem is very complex and lose auditing mechanisms make online attacks scalable and cheap. We observed that a large number of domains in the delivery chain were registered between 1 to 3 days before being used in attacks. This issue is not unique to Covid19-themed attack. The community has been dealing with this issue in other forms of attacks as well. Although domain registrars have considerable authority over the domain registration process, they traditionally tend to remain neutral or act negligently. We observed similar trends when dealing with ASes involved in Covid19-themed attacks. While the more well-known ASes took the issue seriously and stopped hosting those service, many others chose a neutral position about hosting malware and scam domains. We understand that finding the right incentives for registry operators, cloud service providers, and search engines to implement such policies might be challenging on a global scale, as these entities might be wary of policing content on the web. However, a coalition in this direction could be helpful as today browser vendors have become the main or perhaps the only line of defense for protecting Internet users from these attacks.

6.2 Detecting Deceptive Practices

Deceptive practices are occurring pervasively. Our results demonstrate that deceptive patterns are widely used in Covid19-themed attacks. Third-party entities are fueling this trend and actively promoting deceptive practices or even facilitating the process by providing infrastructure to influence consumers. These practices could be used in other forms of web-based social engineering attacks such as survey scams [29], fraudulent shopping websites [25, 50], technical support scams [38]. The forensic data we collected during crawling experiments could be used beyond academic research by consumer protection entities and independent organizations to raise public awareness while identifying unsafe practices at scale. Our forensic data and tool are generalizable and can potentially reduce the cost of locating deceptive practices with minimal human intervention on a large scale. This can be used as a starting point to inform policy and regulation about the type and scale of unsafe practices that emerge abruptly on the web.

Table 8: The top six campaigns and their corresponding infrastructure.

Campaign ID	Size	TLDs	#IPs/#ASs	Top AS/CDN	Top Threats
UA-75488979	239	biz, xyz, site, space, tk, net	206/5	GoDaddy	Scam pages (Treatment Kits, Monetary Assistance)
UA-43126514	103	space, info, net, club	94/4	GoDaddy	AgentTesla, Emotet
UA-116858069	79	xyz, in, info, net, club	44/3	Cloudflare	njRAT, Darkcomet
UA-8406245	75	info, tk, online	34/9	1 and 1 Internet SE	Sodinokibi
UA-163350519	69	info, xyz, org	20/3	GoDaddy	Emotet, AgentTesla
UA-161418004	63	info, net, org	15/2	GoDaddy	Ryuk, Sodinokibi

6.3 Repeated Measurement

Rapid response requires new web measurement tools. There is no lack of evidence that adversaries are agile in re-purposing their attack techniques when a new opportunity would arise. This asymmetry puts defenders at a distinct disadvantage to formulate a rapid and effective response. To change the dynamic of the arms race in a landscape that attacks evolve quite rapidly, it is critical to develop mechanisms that are less sensitive to data shortage, noise, and perturbation. In particular, it is critical to develop tools and techniques that can identify temporal drifts and patient zero threats in a less supervised way. This requires more systemic investigation on what forms of forensics data are still lacking and how the problem can be modeled as an unsupervised or semi-supervised learning problem. Prior work has demonstrated the benefits of longitudinal data to study new attacks in other domains and shed light on previously opaque online attacks [15–17, 21, 22, 42, 46]. If the ability to perform repeated measurements is paired well with unsupervised or semi-supervised techniques, we are more likely to locate trends at the global scale. In fact, the ability to create catalogs of web threats in an unsupervised way can result in building a valuable knowledge base of web threats and their behavior over time, which can be helpful in multiple ways, such as studying the dynamics of emerging web threats, detecting new adversarial practices, and measuring users' exposure to a specific threat.

6.4 Robustness and Generlizability

Our analysis pipeline incorporates multiple vantage points to look at Covid19-themed threats. It is critical to discuss limitations and specific systematic biases that might have had an impact on the robustness of the method and the generalizability of the findings. We discuss four potential risks in running scalable web security experiments similar to ours. First, the initial URL seeds strongly depend on the Google Trends service, which is subjected to several levels of customization, considering geographical locations, language types, temporal sensitivity, and prior search queries by users located in a specific region. For instance, specific trends in the US may not be a popular trend in the European Union. Therefore, the trend keyword list, which form the basis of our data collection, might differ significantly – resulting in a different set of target pages. Second, search engines have to comply with copyright violation laws. Furthermore, they might have put in place extra layers of protection against sensitive topics (e.g., disinformation about the disease) when generating indexed search results, which can have impacts on our collected samples. Our approach to collect malicious samples or scam pages is also a best-effort approach. Since the analysis pipeline was designed to run in an automated manner, minimal levels of interaction were performed with the target websites. Given that many websites tend to deliver content reactively, often based on triggering specific events, the scanning pipeline might not cover results that required more interaction. In general, this study should be viewed as a lower-bound estimate of the Covid19-themed attacks and the corresponding ecosystem.

Third, the scope of this paper is limited to how users might end up visiting a specific social engineering website, and what would be the techniques to defraud victims in the wild. We do not have sufficient data and evidence to measure how effective these attacks are in exfilterating user data compared to other forms of social engineering attacks. More research needs to be done on the socio-technical aspects of these threats to empirically measure when and how these attacks become more successful and ways to educate users to reduce risks of compromise and data breaches during an unprecedented time that we all experienced during Covid19.

Finally, triggering an event can be viewed as a mechanism to exercise new execution paths in JavaScript code. We incorporated some heuristics to partially simulate user interactions (e.g., scrolling down during a visit). However, this does not produce identical user interaction with websites and can be easily detected by visited websites, in this case, scam websites that are known to use cloaking techniques [63]. As one possible future work, we plan to develop a more systematic method to interact with such websites by developing an event listener fuzzer to increase the level of interaction with a visiting website and trigger possible events.

7 RELATED WORK

Covid19-themed Adversarial Operations. The security community studied Covid19-themed threats from different angels. For instance, researchers explained the rise in phishing activities during the pandemic [8, 10] by looking at domain registrations and certificate issuance rates for phishing pages related to Covid19. This paper tries to answer a different question by running a large scale end-to-end analysis of users' exposure to Covid19-themed attacks starting from a web search to attack delivery. The most relevant work to us was done by Pritom et al.[44]. The paper discusses five classes of malicious practices for Covid19-themed attacks by creating a mapping between the attacks and cyber kill chain model. In another work, Pritom et al. [45] proposed a method to identify malicious Covid19-themed websites by extracting whois information, TLD, and lexical features. Xia et al. [60] conduct a study on a particular form of Covid19-themed scam, cryptocurrency scams, where victims were encouraged to make donations via cryptocurrency.

The authors performed an analysis on the delivery mechanism of the scams and how the malicious practices were taken place. Lastly, Lallie et al. [33] performed a timeline analysis of attacks during the pandemic to understand how attacks were crafted and how adversaries used events such as announcements of policies to launch a new attack scenario. Although these research efforts provide novel insights on the attack types, they focus less on actual delivery mechanisms at scale. In particular, our work is different in the sense that we aim to mainly answer how users are exposed to those forms of attacks in their normal browsing session and what would be the chance of getting exposed to a new attack pattern. The output of this paper is in line with the findings of those projects and complements their results about Covid19-themed threats.

Deceptive Practices in Modern Scams. There is a large body of work studying deception in deception on the web ecosystem [7, 12, 32, 47, 48]. In most of these works, the main focus is on the development of attack pages and technology abuse. In most of this work, the main focus is on the attack delivery mechanism. In this paper, we also observed common deceptive techniques used by adversaries to redirect unsuspecting victims into downloading malicious payloads. However, in contrast to existing work, this paper goes one step further and tries to study in more detail how such a user becomes exposed to Covid19-themed attacks on the web, and how victims are tricked and recruited.

Security and Privacy Analysis. There is also a large body of work on the security posture of web users on the Internet. With that said, the goal of our research is in line with the prior work to investigate the prevalence of a new attack such as scareware [26, 28, 62], PUPs [31, 40, 56], and Survey and technical scams [29, 39]. Furthermore, existing research that unveil deceptive practices in a malicious advertisement [9, 20, 35, 51], ad injection [54, 61], crytojacking [27], and other forms of unlawful monetary gain [41, 52] are also related to our work. While parts of the analysis done in this paper also investigate the impact of dynamic features of the web, such as redirection, in the delivery mechanism, our work differs significantly from existing work as Covid19-themed threats have not been explored at scale in depth in prior work.

8 CONCLUSION

In this paper, we conducted an empirical analysis of Covid19-themed attacks. We showed that adversaries updated their attack strategy quite rapidly. The number of Covid19-themed attacks increased 300% in just eight days. The analysis showed that more than 50% of the observed cases were accessible from the Alexa top 60K websites. Our analysis of the delivered payloads also suggests that the most common activities were filesystem exploration, microphone access, and keylogging attempts. The victim's filesystem was explored in 70% of all cases. The prevalence of actions that attempt to collect information about users and their files suggests that surveillance was a dominant goal in Covid19-themed attacks. Finally, we provide perspectives on how we might respond more effectively to such events in the future.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their thoughtful feedback. This work was supported by Microsoft Security research.

REFERENCES

[1] 2020. Google Trends. https://trends.google.com/trends. [Online; accessed 28. Aug. 2022].
[2] 2020. UK sees a 31% increase in cyber crime amid the pandemic. https://www.securitymagazine.com/articles/93722-uk-sees-a-31-increase-in-cyber-crime-amid-the-pandemic. [Online; accessed 27. Aug. 2022].
[3] 2021. Cybersecurity Tips for Tax Season. https://amtrustfinancial.com/blog/small-business/cybersecurity-tips-for-tax-season. [Online; accessed 28. Aug. 2022].
[4] 2021. See Which States and Cities Have Told Residents to Stay at Home. https://www.nytimes.com/interactive/2020/us/coronavirus-stay-at-home-order.html. [Online; accessed 28. Aug. 2022].
[5] 2022. How Cybercrime Evolved Through The Pandemic: Year by Year. https://www.globalbrandsmagazine.com/how-cybercrime-evolved-through-the-pandemic-year-by-year/. [Online; accessed 27. Aug. 2022].
[6] 2022. Phishtank:Fighting against phishing. https://phishtank.org/.
[7] Sherly Abraham and InduShobha Chengalur-Smith. 2010. An overview of social engineering malware: Trends, tactics, and implications. *Technology in Society* 32, 3 (2010), 183–196. https://doi.org/10.1016/j.techsoc.2010.07.001
[8] Ali Al-Qahtani and Stefano Cresci. 2022. The COVID-19 scamdemic: A survey of phishing attacks and their countermeasures during COVID-19. *IET Information Security* 16 (09 2022), n/a–n/a. https://doi.org/10.1049/ise2.12073
[9] Sajjad Arshad, Amin Kharraz, and William Robertson. 2016. Include Me Out: In-Browser Detection of Malicious Third-Party Content Inclusions. In *Proceedings of the 20th International Conference on Financial Cryptography and Data Security (FC)* (Barbados).
[10] Marzieh Bitaab, Haehyun Cho, Adam Oest, Penghui Zhang, Zhibo Sun, Rana Pourmohamad, Doowon Kim, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupe, and Gail Joon Ahn. 2020. Scam Pandemic: How Attackers Exploit Public Fear through Phishing. In *Proceedings of the 2020 APWG Symposium on Electronic Crime Research, eCrime 2020 (eCrime Researchers Summit, eCrime)*. IEEE Computer Society. Publisher Copyright: © 2020 IEEE.; 2020 APWG Symposium on Electronic Crime Research, eCrime 2020 ; Conference date: 16-11-2020 Through 19-11-2020.
[11] Bradley Barth. 2020. Open redirect on Dept. of HHS website benefits COVID-19 phishing scam. https://www.scmagazine.com/home/security-news/cybercrime/open-redirect-on-dept-of-hhs-website-benefits-covid-19-phishing-scam/.
[12] Juan Caballero, Chris Grier, Christian Kreibich, and Vern Paxson. 2011. Measuring Pay-per-Install: The Commoditization of Malware Distribution. In *20th USENIX Security Symposium (USENIX Security 11)*. USENIX Association, San Francisco, CA. https://www.usenix.org/conference/usenix-security-11/measuring-pay-install-commoditization-malware-distribution
[13] Catalin Cimpanu. 2020. State-sponsored hackers are now using coronavirus lures to infect their targets. State-sponsoredhackersarenowusingcoronaviruslurestoinfecttheirtargets.
[14] Covenant HealthCovenant Health. 2020. World Health Organization Warns of Coronavirus (COVID-19) Phishing Attacks. https://covenanthealth.com/world-health-organization-who-warns-of-coronavirus-covid-19-phishing-attacks/.
[15] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. Association for Computing Machinery, New York, NY, USA, 542–553. https://doi.org/10.1145/2810103.2813703
[16] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, and J. Alex Halderman. 2014. The Matter of Heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC '14)*. Association for Computing Machinery, New York, NY, USA, 475–488. https://doi.org/10.1145/2663716.2663755
[17] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-Wide Scanning and Its Security Applications. In *Proceedings of the 22nd USENIX Conference on Security* (Washington, D.C.) *(SEC'13)*. USENIX Association, USA, 605–620.
[18] Google Development. 2017. Extending DevTools. https://developer.chrome.com/extensions/devtools.
[19] RUESE GROUP. 2022. Be Alert for Cyber Crime during Tax Season. https://rueseinsurancegroup.com/be-alert-for-cyber-crime-during-tax-season/. [Online; accessed 28. Aug. 2022].
[20] Hamed Haddadi. 2010. Fighting online click-fraud using bluff ads. *ACM SIGCOMM Computer Communication Review* 40, 2 (2010), 21–25.
[21] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. 2008. Census and survey of the visible internet (extended). Technical Report ISI-TR-2008-649b. (2008).

[22] Ralph Holz, Lothar Braun, Nils Kammenhuber, and Georg Carle. 2011. The SSL Landscape: A Thorough Analysis of the x.509 PKI Using Active and Passive Measurements. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference* (Berlin, Germany) *(IMC '11)*. Association for Computing Machinery, New York, NY, USA, 427–444. https://doi.org/10.1145/2068816.2068856

[23] Piotr Indyk and Rajeev Motwani. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (Dallas, Texas, USA) *(STOC '98)*. Association for Computing Machinery, New York, NY, USA, 604–613. https://doi.org/10.1145/276698.276876

[24] Jessica Davis. 2020. Another COVID-19 Research Firm Targeted by Ransomware Attack. https://healthitsecurity.com/news/another-covid-19-research-firm-targeted-by-ransomware-attack.

[25] Dr Khan and W Shazia. 2019. Cyber security issues and challenges in E-commerce. In *Proceedings of 10th international conference on digital strategies for organizational success*.

[26] Amin Kharraz, Sajjad Arshad, Collin Mulliner, William Robertson, and Engin Kirda. 2016. UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, 757–772.

[27] Amin Kharraz, Zane Ma, Paul Murley, Charles Lever, Joshua Mason, Andrew Miller, Manos Antonakakis, and Michael Bailey. 2019. Outguard: Detecting In-Browser Covert Cryptocurrency Mining in the Wild. In *The Proceedings of the 2019 World Wide Web Conference (WWW '19)*. San Francisco, CA.

[28] Amin Kharraz, William Robertson, Davide Balzarotti, Leyla Bilge, and Engin Kirda. 2015. Cutting the gordian knot: A look under the hood of ransomware attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 3–24.

[29] Amin Kharraz, William Robertson, and Engin Kirda. 2018. Surveylance: Automatically Detecting Online Survey Scams. In *2018 IEEE Symposium on Security and Privacy (SP)*. 70–86.

[30] Platon Kotzias, Leyla Bilge, and Juan Caballero. 2016. Measuring PUP Prevalence and PUP Distribution through Pay-Per-Install Services. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kotzias. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 739–756.

[31] Platon Kotzias, Leyla Bilge, and Juan Caballero. 2016. Measuring PUP Prevalence and PUP Distribution through Pay-Per-Install Services.. In *USENIX Security Symposium*. 739–756.

[32] Bum Jun Kwon, Jayanta Mondal, Jiyong Jang, Leyla Bilge, and Tudor Dumitraş. 2015. The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. Association for Computing Machinery, New York, NY, USA, 1118–1129. https://doi.org/10.1145/2810103.2813724

[33] Harjinder Singh Lallie, Lynsay A. Shepherd, Jason R.C. Nurse, Arnau Erola, Gregory Epiphaniou, Carsten Maple, and Xavier Bellekens. 2021. Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic. *Computers & Security* 105 (jun 2021), 102248. https://doi.org/10.1016/j.cose.2021.102248

[34] John Leyden. 2021. Ransomware attacks more than doubled last year as cybercrime operations scale up during coronavirus pandemic. https://portswigger.net/daily-swig/ransomware-attacks-more-than-doubled-last-year-as-cybercrime-operations-scale-up-during-coronavirus-pandemic. [Online; accessed 27. Aug. 2022].

[35] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2012. Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 674–686.

[36] Paul Mee and Rico Brandenburg. 2020. CYBER RISK GROWS AS CRIMINALS EXPLOIT CORONAVIRUS CRISIS. https://www.oliverwyman.com/our-expertise/insights/2020/apr/risk-journal-vol-9/rethinking-tactics/cyber-risk-grows-as-criminals-exploit-coronavirus-crisis.html. [Online; accessed 27. Aug. 2022].

[37] Microsoft Corporation. 2017. Cognitive Services Pricing – Bing Search API. https://azure.microsoft.com/en-us/pricing/details/cognitive-services/search-api/web/.

[38] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. 2016. Dial one for scam: A large-scale analysis of technical support scams. *arXiv preprint arXiv:1607.06891* (2016).

[39] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. 2016. Dial One for Scam: Analyzing and Detecting Technical Support Scams. *CoRR* abs/1607.06891 (2016). arXiv:1607.06891 http://arxiv.org/abs/1607.06891

[40] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. 2016. Towards Measuring and Mitigating Social Engineering Software Download Attacks. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 773–789.

[41] Nick Nikiforakis, Federico Maggi, Gianluca Stringhini, M Zubair Rafique, Wouter Joosen, Christopher Kruegel, Frank Piessens, Giovanni Vigna, and Stefano Zanero. 2014. Stranger danger: exploring the ecosystem of ad-based url shortening services. In *Proceedings of the 23rd international conference on World wide web*. ACM, 51–62.

[42] Tu Ouyang, Soumya Ray, Mark Allman, and Michael Rabinovich. 2014. A Large-Scale Empirical Analysis of Email Spam Detection through Network Characteristics in a Stand-Alone Enterprise. *Computer Networks* 59 (2014), 101–121. https://doi.org/10.1016/j.comnet.2013.08.031

[43] DAN PATTERSON. 2021. Cybercrime is thriving during the pandemic, driven by surge in phishing and ransomware. https://www.cbsnews.com/news/ransomware-phishing-cybercrime-pandemic/. [Online; accessed 27. Aug. 2022].

[44] Mir Mehedi Ahsan Pritom, Kristin M. Schweitzer, Raymond M. Bateman, Min Xu, and Shouhuai Xu. 2020. Characterizing the Landscape of COVID-19 Themed Cyberattacks and Defenses. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE. https://doi.org/10.1109/isi49825.2020.9280539

[45] Mir Mehedi Ahsan Pritom, Kristin M. Schweitzer, Raymond M. Bateman, Min Xu, and Shouhuai Xu. 2020. Data-Driven Characterization and Detection of COVID-19 Themed Malicious Websites. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE. https://doi.org/10.1109/isi49825.2020.9280522

[46] Niels Provos and Peter Honeyman. 2001. ScanSSH: Scanning the Internet for SSH Servers. In *Proceedings of the 15th Conference on Systems Administration (LISA 2001), San Diego, California, USA, December 2-7, 2001*. 25–30.

[47] Babak Rahbarinia, Marco Balduzzi, and Roberto Perdisci. 2016. Real-Time Detection of Malware Downloads via Large-Scale URL->File->Machine Graph Mining. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security* (Xi'an, China) *(ASIA CCS '16)*. Association for Computing Machinery, New York, NY, USA, 783–794. https://doi.org/10.1145/2897845.2897918

[48] Fatima Salahdine and Naima Kaabouch. 2019. Social engineering attacks: A survey. *Future Internet* 11, 4 (2019), 89.

[49] VirusTotal Online Service. [n. d.]. VirusTotal. https://www.virustotal.com/gui/home.

[50] Nashrudin Setiawan, Vita Cita Emia Tarigan, Pipit Buana Sari, Yossie Rossanty, MDTP Nasution, and Ilhamsyah Siregar. 2018. Impact Of Cybercrime In E-Business And Trust. *Int. J. Civ. Eng. Technol* 9, 7 (2018), 652–656.

[51] Kevin Springborn and Paul Barford. 2013. Impression Fraud in On-line Advertising via Pay-Per-View Networks.. In *USENIX Security*. 211–226.

[52] Brett Stone-Gross, Ryan Abman, Richard A Kemmerer, Christopher Kruegel, Douglas G Steigerwald, and Giovanni Vigna. 2013. The underground economy of fake antivirus software. In *Economics of Information Security and Privacy III*. Springer, 55–78.

[53] The Hacker News. 2020. Beware of 'Coronavirus Maps' – It's a malware infecting PCs to steal passwords. https://thehackernews.com/2020/03/coronavirus-maps-covid-19.html.

[54] Kurt Thomas, Elie Bursztein, Chris Grier, Grant Ho, Nav Jagpal, Alexandros Kapravelos, Damon McCoy, Antonio Nappa, Vern Paxson, Paul Pearce, et al. 2015. Ad injection at scale: Assessing deceptive advertisement modifications. In *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 151–167.

[55] Kurt Thomas, Juan A Elices Crespo, Ryan Rasti, Jean Michel Picod, Cait Phillips, Marc-André Decoste, Chris Sharp, Fabio Tirelo, Ali Tofigh, Marc-Antoine Courteau, et al. 2016. Investigating Commercial Pay-Per-Install and the Distribution of Unwanted Software.. In *USENIX Security Symposium*. 721–739.

[56] Kurt Thomas, Juan A. Elices Crespo, Ryan Rasti, Jean-Michel Picod, Cait Phillips, Marc-André Decoste, Chris Sharp, Fabio Tirelo, Ali Tofigh, Marc-Antoine Courteau, Lucas Ballard, Robert Shield, Nav Jagpal, Moheeb Abu Rajab, Panayiotis Mavrommatis, Niels Provos, Elie Bursztein, and Damon McCoy. 2016. Investigating Commercial Pay-Per-Install and the Distribution of Unwanted Software. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 721–739.

[57] ThreatMicro Inc. 2020. Emotet Uses Coronavirus Scare in Latest Campaign, Targets Japan. https://www.trendmicro.com/vinfo/mx/security/news/cybercrime-and-digital-threats/emotet-uses-coronavirus-scare-in-latest-campaign-targets-japan.

[58] P Upatham and J Treinen. 2020. Amid covid-19, global orgs see a 148% spike in ransomware attacks; finance industry heavily targeted. https://blogs.vmware.com/security/2020/04/amid-covid-19-global-orgs-see-a-148-spike-in-ransomware-attacks-finance-industry-heavily-targeted.html. [Online; accessed 27. Aug. 2022].

[59] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *Image Processing, IEEE Transactions on* 13 (05 2004), 600 – 612. https://doi.org/10.1109/TIP.2003.819861

[60] Pengcheng Xia, Haoyu Wang, Xiapu Luo, Lei Wu, Yajin Zhou, Guangdong Bai, Guoai Xu, Gang Huang, and Xuanzhe Liu. 2020. Don't Fish in Troubled Waters! Characterizing Coronavirus-themed Cryptocurrency Scams. In *2020 APWG Symposium on Electronic Crime Research (eCrime)*. 1–14. https://doi.org/10.1109/eCrime51433.2020.9493255

[61] Xinyu Xing, Wei Meng, Byoungyoung Lee, Udi Weinsberg, Anmol Sheth, Roberto Perdisci, and Wenke Lee. 2015. Understanding Malvertising Through Ad-Injecting Browser Extensions. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*.

[62] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. 2014. The dark alleys of madison avenue: Understanding malicious advertisements. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 373–380.

[63] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, et al. 2021. Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1109–1124.

A APPENDIX

Table 9: Timeline of Events in Late January to Mid April 2020. Several targeted attacks on Bio technology research firms, government entities, and health organizations raised concerns in March. In April, we witnessed an increase in large-scale attacks against end-users using the ongoing Covid19 pandemic as a hook to hoodwink victims into running malware.

Date	Incidents
01/20/2020	Japanese entities were targeted by Covid19 Emotet spam campaigns. [57]
03/02/2020	10x Genomics Inc., a Covid19 research firm, experienced an attempted ransomware attack and data breach [24].
03/03/2020	Rogue executables masquerade legitimate Covid19 map from Johns Hopkins University. [53]
03/10/2020	Scam campaigns impersonated WHO requesting Bitcoin donations [14].
02/27/2020	Covid19-themed spam campaigns targeted users in Ukraine, China, Italy, Spain disguising as updates from the 'Center for Public Health'
03/13/2020	Nation backed hackers Mustang Panda (China) utilized multiple scam campaigns to deliver malicious code [13]
03/24/2020	The US Health and Human Services website (hhs.gov) was redirecting to a malicious infrastructure distributing Racoon [11]
03/18/2020	Android malware distributed ransomware via Covid19
04/09/2020	FBI public service announcement about the increased cyber threats surrounding COVID19 pandemic
⋮	⋮
04/17/2020	Google blocked 18 million daily malware and phishing emails related to Covid19 in addition to 240 million Covid19-related daily spam messages.

Table 10: A List of Windows APIs that can be used to automatically launch additional payloads.

API	Description
CreateProcess()	Creates a new process and its primary thread.
CreateProcessAsUserA()	Creates a new process with the security context of the given user.
CreateProcessWithLogonW()	Creates a new process in the security context of the specified user credentials.
LoadLibraryA()	Loads the specified module into the address space of the calling process.
ShellExecuteA()	Performs an operation on a specified file.

Table 11: The list of malware families used for cataloging.

Training Data (Malicious)	Examples	Unique MD5s
Ransomware	Sodinokibi, Ryuk, Locky, PoshCoder, TeslaCrypt	110
Trojan	NanoCore, Racoon, Qbot, njRAT	190
Total	-	300

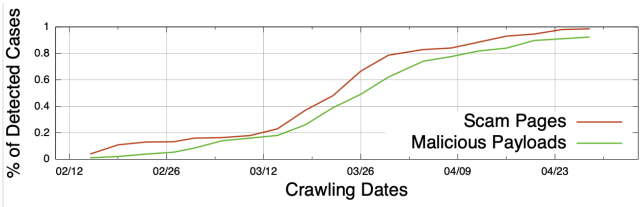

Figure 6: The upsurge in the number of Covid19-themed attacks. We noticed an abrupt increase in scam pages as well as malware families after March 15th.

Table 12: Top third-party ad providers that delivered Covid19-themed attack pages.

Rank	Third-party Entity	(#)
1	spotxchange.com	13.3%
2	adcash.com	10%
3	doubleclick.net	9%
4	direct-yandex.com	11%
5	viglink.com	6%
6	adroll.com	6.3%
7	onlickads.net	5.8%
8	adify.com	3.4%
9	prebig.org	2.7%

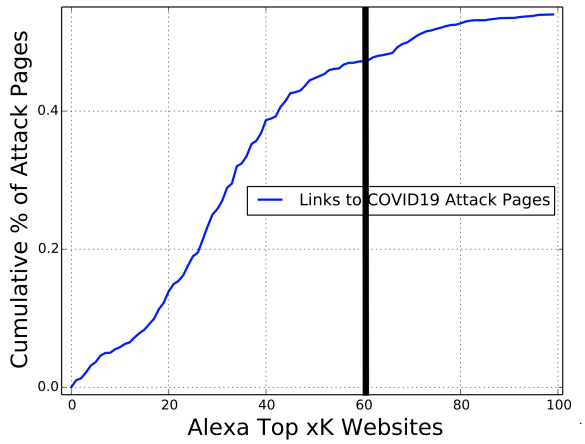

Figure 7: The reachability of survey gateways from top websites. 50% of the Covid19-themed attacks were reachable from the Alexa top 60K websites.

Table 13: The most common hosting locations of Covid19-themed attack delivery servers.

No.	Country	Covid19-themed
1	France	54
2	Germany	294
3	India	55
4	Japan	29
5	Netherlands	132
6	Poland	39
7	Russia	119
8	Spain	29
9	United Kingdom	110
10	US	1,715