IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 April 2023, accepted 19 May 2023, date of publication 29 May 2023, date of current version 5 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280815

RESEARCH ARTICLE

Stargazer: Long-Term and Multiregional Measurement of Timing/ Geolocation-Based Cloaking

SHOTA FUJII[1,2], TAKAYUKI SATO[1], SHO AOKI[1], YU TSUDA[3], NOBUTAKA KAWAGUCHI[1], TOMOHIRO SHIGEMOTO[1], AND MASATO TERADA[1]

[1]Research and Development Group, Hitachi Ltd., Yokohama, Kanagawa 244-0817, Japan
[2]Graduate School of Natural Science and Technology, Okayama University, Okayama 700-8530, Japan
[3]National Institute of Information and Communications Technology, Tokyo 184-8795, Japan

Corresponding author: Shota Fujii (shota.fujii.xh@hitachi.com)

This work was supported by the National Institute of Information and Communications Technology (NICT) Research Theme "Cyber-Attacks Data Collection and Analysis for Elucidation of Targeted Attack Methods."

ABSTRACT Malicious hosts have come to play a significant and varied role in today's cyber attacks. Some of these hosts are equipped with a technique called cloaking, which discriminates between access from potential victims and others and then returns malicious content only to potential victims. This is a serious threat because it can evade detection by security vendors and researchers and cause serious damage. As such, cloaking is being extensively investigated, especially for phishing sites. We are currently engaged in a long-term cloaking study of a broader range of threats. In the present study, we implemented Stargazer, which actively monitors malicious hosts and detects geographic and temporal cloaking, and collected 30,359,410 observations between November 2019 and February 2022 for 18,397 targets from 13 sites where our sensors are installed. Our analysis confirmed that cloaking techniques are widely abused, i.e., not only in the context of specific threats such as phishing. This includes geographic and time-based cloaking, which is difficult to detect with single-site or one-shot observations. Furthermore, we found that malicious hosts that perform cloaking include those that survive for relatively long periods of time, and those whose contents are not present in VirusTotal. This suggests that it is not easy to observe and analyze the cloaking malicious hosts with existing technologies. The results of this study have deepened our understanding of various types of cloaking, including geographic and temporal ones, and will help in the development of future cloaking detection methods.

INDEX TERMS Cloaking, cyber security, geofencing, malcious host, time-series.

I. INTRODUCTION

Today's cyber attacks utilize various types of malicious hosts to carry out attacks. For example, there are cases in which files on infected terminals are uploaded to external servers using Hypertext Transfer Protocol (HTTP) communication [1], and others where attack modules are downloaded or attack commands are received [2]. In addition to use as C2 servers, malicious hosts have various other roles such as malware distribution [3], [4], phishing [5], [6], [7], fraudulent services [8], [9] spam distribution [10], and fake Anti-Virus (AV) scanner distribution [11], [12]. The number of these malicious hosts has been increasing exponentially year by year [13]. In such circumstances, it is important to block communications to suspicious servers used for cyber attacks in order to limit the damage.

In response, attackers have implemented evasion techniques against malicious host detection techniques. Such evasion techniques are known as cloaking. Cloaking determines whether the access is from the target organization or region and then returns benign content if not, i.e., if the access is from outside the target, including security researchers.

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

Such techniques bypass security researcher investigations and automatic malicious host detection systems, resulting in threat information not being shared and action being delayed. For example, an extensive study of phishing in [14] examined the possibility that there may be some attacks that cannot be observed due to cloaking.

Several prior studies have investigated the characteristics of cloaking, especially those related to phishing, ad network exploitation, fake AV distribution, etc. [5], [6], [7], [15], [16], [17]. On the other hand, cloaking for non-human-mediated items, which differs from these methods, is less well understood. Websites using evasion techniques for these attacks achieve cloaking mainly by distinguishing between automated crawler visits and potential human victims to determine whether they are attack targets or not [15]. However, other communications in a generic attack flow, such as communication with a C2 server or downloading an attack module, do not involve human intervention, and thus the distinction between attack targets and others does not necessarily correspond to the distinction between humans and crawlers. Examples include cloaking based on the country or IP address of the access source or short-term enabling of C2 servers. For example, a study examining malware distribution via Pay-Per-Install (PPI) showed that some malware families are biased toward geographic areas of distribution [18]. Another study examining malware distributed via PPI from U.S. IP addresses demonstrated that results may differ when observed from non-U.S. IP addresses [19]. There are also reports that some malware families and exploit kits are distributed or C2 servers are enabled only in specific regions [20], [21], [22]. Therefore, assuming that cloaking is based on the discrimination of crawlers or humans, as suggested by the results of previous studies, may lead to overlooking malicious hosts. This results in the risk of serious damage to the organization or region targeted by the cloaking.

Although the several existing studies discussed above have pointed out the importance of defenses against cloaking, we still lack knowledge on the long-term and regional tendencies of malware distribution and C2 servers, which makes it difficult to implement efficient and effective countermeasures. To address this issue, we conducted a measurement study of temporal and regional cloaking, focusing primarily on malware distribution hosts and C2 servers. This was done by implementing Stargazer, a platform that enables multiple sensors deployed around the world to observe simultaneously and in parallel over a long period of time, avoiding temporal and regional cloaking (known as *geofencing*). We utilized Stargazer by extending the preliminary experiments (continuous observation using monitoring sensors from around the world) conducted in [23]. Specifically, we implemented a mechanism to automatically detect cloaking from observation results, and also increased the number of observations and observation sensors to conduct large-scale observation experiments, analysis, and discussion.

In the observation process, Stargazer keeps periodically accessing a set of suspicious hosts from sensors installed in several different regions to analyze how the temporal and regional differences of the access origins affect the characteristics and contents of the responses from the hosts. The specific observation items are executing HTTP GET, obtaining Domain Name System (DNS) records, obtaining a screenshot, and sending a ping. We also used Stargazer to conduct observations from November 2019 to February 2022 on 18,397 targets from 13 locations, and obtained 30,359,410 observations. We analyzed these observations and found that approximately 15% of the suspicious hosts were attempting geofencing, and approximately 17% were attempting time-based cloaking. Our analysis confirms that cloaking techniques are widely exploited. It includes geofencing and time-based cloaking, which is difficult to detect with single-site or one-shot observations.

In addition, we analyzed observations from hosts labeled as cloaking to determine the details of cloaking. Specifically, we systematically revealed the existence of seven different methods: three types of time-based cloaking and four types of geofencing. Furthermore, we confirmed that the malicious hosts for cloaking include those that survive for a relatively long period of time by using their detection evasion ability as a shield and those whose contents do not exist in VirusTotal. This suggests that it is not easy to observe and analyze the cloaking malicious hosts using existing technologies, and that they are a threat not only to general users but also to researchers.

These analyses have advanced our understanding of geofencing and time-based cloaking of malicious hosts, which had previously been discussed only qualitatively. We believe that Stargazer not only automatically reveals geofencing and time-based cloaking, thus making it more difficult for attackers to evade detection, but also contributes to the development of future cloaking detection methods. In summary, the contributions of this paper are as follows:

- To conduct a measurement, we designed and implemented Stargazer, which enables time-series analysis and regional analysis of malicious hosts. Stargazer observes malicious hosts from multiple geolocations, which makes it difficult for adversaries to evade observation through cloaking (§III).
- We conducted observations by Stargazer from November 2019 to February 2022 for 18,397 targets from 13 locations, and obtained 30,359,410 observation results. We quantitively analyzed the observation results and confirmed that cloaking is omnipresent among malicious hosts related to various threats. Our measurement study also clarified that there are seven major types of cloaking types: three types of time-based cloaking and four types of geofencing. We also found that cloaked content had significantly fewer submissions to reputation sites such as VirusTotal than those that were not cloaked (§IV).

- We qualitatively deep-dived into the more characteristic observations as case studies. We showed a host that survived for a long time in total by restarting after being inactive once, and a host that attempted to avoid detection by frequently changing its IP address and contents in a short period of time. We also showed that among the cloaking sites of the same campaign, the malware distribution site, which can be easily detected, has a relatively short survival time and is disposable, while the C2 server in the latter stage has a long survival time and is reused, and thus has different characteristics depending on the host role (§V).

II. BACKGROUND

A. CLOAKING TECHNIQUES IN CYBER ATTACKS

Some Internet services change the content they provide according to the visitors to the site. For example, some systems identify the visitor's device or browser and return content for mobile devices and PCs, while others identify the visitor's region and change the language of the content accordingly. These are benign, taking into account the improvement of usability. On the other hand, these identification techniques have been exploited in cyber attacks, as mentioned above, by cloaking.

A typical cloaking scheme identifies whether a visitor is a bot or not and returns benign content for bots and malicious content for people. Others return malicious content only to the target region or to targets that may contain vulnerabilities to be exploited. This enables the distribution of malicious content only to the right target, and delays or avoids detection by automatic detection systems or security researchers.

In summary, existing research often focuses on specific categories of attacks, especially phishing. In addition, these studies are tailored methods that utilize features specific to hosts with web pages, including cloaking sites, such as screenshot differences and JavaSript structures. On the other hand, there are reports of cloaking being performed on malicious hosts that are not limited to these attacks. Thus, although there is evidence that cloaking is universally utilized in a wide range of attack categories, prior research has conducted few large-scale investigations.

B. CHALLENGES IN CLOAKING OBSERVATION

With the above background, many cloaking detection methods have been proposed and cloaking methods have been investigated. In particular, cloaking tends to be implemented for phishing sites because of the conflicting demands of the search engines to rank them higher through Search Engine Optimization in order to reach more attack targets, while at the same time not wanting them to be analyzed by researchers. Thus, research on this issue is also active [15], [16], [24], [25].

On the other hand, cloaking has also been conducted on malicious hosts that are not limited to these types of threats. One survey implementing manual analysis [18] showed that some families and types of malware have regional characteristics. Specifically, a technique called geofencing is sometimes used to evade detection by distributing malicious content only to the target region and returning harmless content or not responding to accesses from other regions. There are also reports of geofencing being used in *Sharkbot* [20], *Gamaredon* [21], and *Purple Fox* [22] to target specific regions. Cloaking using browser user agents has also been reported [26].

Malicious hosts do not always continue to behave in a malicious manner; they may be activated only at the time of an attack, or they may become temporarily dormant, or they may be permanently destroyed. In particular, cloaking that is activated only when an attack is to be carried out is referred to as timing-based cloaking in this paper. Many existing studies have only observed each host at a certain point in time, and the changes over time and the cloaking within them have not been clarified.

III. METHOD

A. OVERVIEW

This section introduces Stargazer, a system that actively monitors malicious hosts to obtain an overall understanding of their characteristics. The objective of Stargazer in the present study is to investigate the tendencies of the geofencing and time-based cloaking mentioned in the previous section. First, to enable detection of time-based cloaking, observations of malicious hosts are made periodically and continuously, rather than once. In this case, the same site is managed with a unique ID and maintained in a form that allows time series analysis for each site. In addition, to enable the detection of geofencing, observation sensors that observe malicious hosts are installed in multiple regions. This improves the observability of hosts that return malicious responses only to accesses from specific regions.

Fig. 1 shows an overview of Stargazer based on the above. It consists of three components: an observation server, observation sensors, and an analysis system, and executes observation and analysis of malicious hosts by the following processes.

1) The observation server orders observation sensors to start monitoring with the targets' URLs periodically.
2) Observation sensors monitor the targets' URLs on the basis of the order.
3) After the observation is completed, the observation results are analyzed using the analysis system.
4) The results of observation and analysis are stored in the observation server's database.

Each component is described in detail in the following sections.

B. OBSERVATION SERVER

The observation server is the command center. It sends URLs to be periodically and continuously observed to observation sensors installed all over the world and then orchestrates

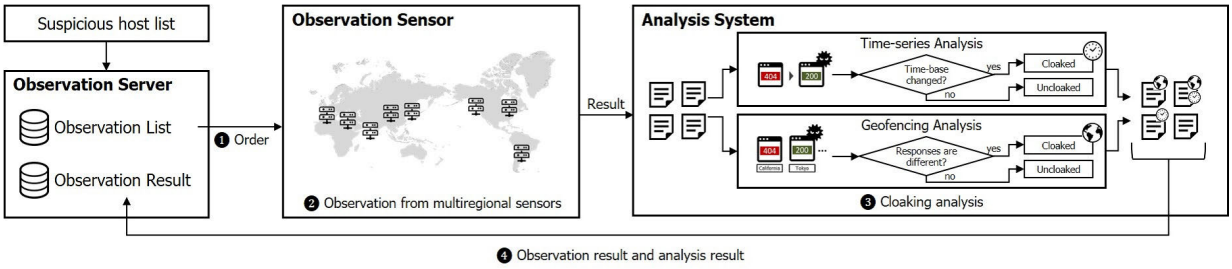


FIGURE 1. Overview of stargazer.

the sensors. After sensor observation is completed for each period, the server receives the results of the observation and analysis and saves them in its database. The periodic and continuous observation enables us to detect time-series changes of malicious hosts and to obtain changes in their status.

C. OBSERVATION SENSORS

The observation sensor monitors the malicious host on the basis of the observation command from the observation server. The specific observation items are as follows:

- Executing HTTP GET
- Obtaining A/AAAA records
- Obtaining a screenshot
- Sending a ping

As mentioned above, HTTP communication is one of the main communication methods used in attacks. By executing HTTP GET for each destination, the status code and contents of each malicious host can be obtained. In addition, the A/AAAA record is obtained as an HTTP GET-related item, and a screenshot is obtained using a headless browser. A record is an IPv4 address assigned to the target host, such as *192.0.2.1*. The AAAA record is an IPv6 address assigned to the target host, such as *2001:db8::1*. Pings are then sent to check the operational status of the adversaries' server. Furthermore, DNS records are obtained at irregular intervals to determine whether domain is sinkholed (sinkhole determination method is detailed in section IV-A).

The observation results are then sent to the observation server. If a redirect is detected, the same observation will be executed recursively for the redirected page.

To make observation sensors resistant against geofencing, we deployed them in multiple geolocations and conducted observations from each sensor simultaneously and in parallel. We set up the observation sensors in 13 geolocations with the motivation of covering a wide area. Twelve of the sensors were installed at each geolocation of Amazon Web Services (AWS), and the remaining sensor was installed on-premise in Japan. Table 1 shows the breakdown of the installation geolocations and platforms (PFs). We confirmed the IP address of each sensor was correctly associated with the geolocation shown in Table 1 by using the GeoIP2 database.[1] In this experiment, the observation sensor is implemented as a Python program. The on-premises sensor runs on Ubuntu 18.04 LTS VM on ESXi 6.7, and the AWS sensor runs on Ubuntu 18.04 LTS VM on Amazon Elastic Compute Cloud (Amazon EC2).

[1] https://www.maxmind.com/en/geoip2-databases

TABLE 1. Location of observation sensors and platforms (PFs).

No.	Geolocations	PF
1	US West (N. California)	AWS
2	US East (N. Virginia)	AWS
3	Europe (Frankfurt)	AWS
4	Europe (London)	AWS
5	Europe (Milan)	AWS
6	Middle East (Bahrain)	AWS
7	South America (Sao Paulo)	AWS
8	Asia Pacific (Hong Kong)	AWS
9	Asia Pacific (Mumbai)	AWS
10	Asia Pacific (Singapore)	AWS
11	Asia Pacific (Sydney)	AWS
12	Asia Pacific (Tokyo)	AWS
13	Japan	on-premise

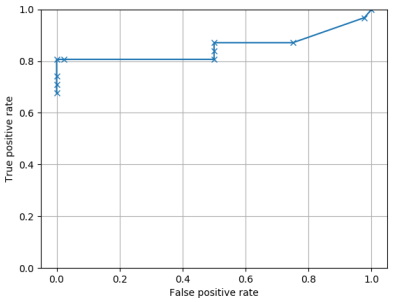
D. ANALYSIS SYSTEM

The analysis system executes time series and geofencing analyses on the observed data. Each analysis item is described in the following sections.

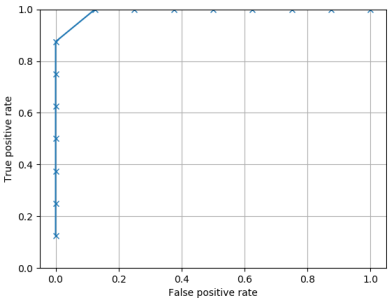
1) TIME-SERIES ANALYSIS

In the time-series analysis, the change rate from time *t-1* to time *t* at the same observation sensor is calculated using *formula 1*, and if the change rate is higher than a threshold, it is extracted as a change in the time series. *Formula 1* was inspired by the Jaccard index, which measures the similarity of sets. It takes the set similarity of the observation results S and detects no change when the similarity is high (i.e., there is no difference in the observation results) and detects change when the similarity is low (i.e., there is a difference). S consists of the observation items described in Section III-C. For example, when the status code is 200, the content is html, and the response to the ping is yes, as $S = \{200, html, ping : yes\}$ (other factors such as the hash value of the content are also included). S_t then means the observation result observed at time t in this measurement. After forming a set of discrete values at each time, the similarity of the sets is calculated to determine whether the response has changed over time.

There are two types of change regarding the content: small, such as a site that contains variable information (e.g., time information), and large, such as a change from a harmless



(A) time-based cloaking detection.



(B) geofencing detection.

FIGURE 2. ROC curves to detect cloakings.

file to malware. If we treat both changes as the same, we may induce false positives caused by the small ones. Therefore, for content, instead of using discrete values of whether the hash values are different, we use fuzzy hashing to calculate the similarity between content as a continuous value and integrate it into the aggregate similarity. The fuzzy hashes are used to calculate the similarity between content in continuous values and integrated into the set similarity. The set similarity of the content of *t-1* and the content of *t* is defined as $Sim(Content(t-1), Content(t))$ in *formula 1*. We utilized *ssdeep*[2] as a fuzzy hash, which can calculate the similarity between two contents as a continuous value. By using it, the aforementioned file similarity can be incorporated into *formula 1*.

$$change\_rate(t-1,t) = 1 - \left(\frac{S_{t-1} \wedge S_t}{S_{t-1} \vee S_t} * Sim(Content(t-1), Content(t))\right) \quad (1)$$

2) GEOFENCING ANALYSIS

In geofencing analysis, the difference between the observation results of sensors s_1 to s_n at the same time is calculated using *formula 2*, and if the difference is higher than a threshold, the sensor is considered to have geofencing characteristics, i.e., cloaking is executed in accordance with the access source. For example, s_1 means a sensor located at US West (AWS) and S_{s_1} means an observation result of s_1. *Formula 2* takes the set similarity of S and detects that there is no regionality when the set similarity is high, and that it is geofenced when the set similarity is low. For content, similarity is calculated as a discrete value using *ssdeep*, the same as for time-series changes. In geofensing analysis, a site that changes its language depending on the access source may be judged as a cloaking even if its change is minor and it is not malicious. Thus, we utilize *ssdeep* as fuzzy hash to suppress false positives.

$$geofenced\_rate(s_1,\ldots,s_n) = 1 - \left(\frac{S_{s_1} \wedge \ldots \wedge Ss_n}{S_{s_1} \vee \ldots \vee S_{s_n}} * Sim(Content(s_1), \ldots, Content(s_n))\right) \quad (2)$$

[2]https://ssdeep-project.github.io/ssdeep/

E. PRELIMINARY EVALUATION

In this section, we evaluate the accuracy of formulas 1 and 2 described in section III-D and attempt to determine the threshold for each formula. For the evaluation, we constructed a dataset by extracting from a subset of Stargazer's observations those of malicious hosts for which manual geofencing and time-based cloaking were performed. Specifically, we checked whether the observation results of the cloaking and non-cloaking targets were malicious and benign, respectively, and confirmed that the cloaking seems to be malicious and targeted. The benign observation results include simple error pages and default web server pages such as Apache, and the malicious observation results include pages that contain malicious code or pages that download malware.

We then applied formulas 1 and 2 to the data set, calculated the time-series change and geofencing scores, and created Receiver Operating Characteristic (ROC) curves. The results are shown in Fig. 2. We selected the thresholds of 0.35 (true positive rate: 81.2%, false positive rate: 5.0%) for time-based cloaking and 0.23 (true positive rate: 92.1%, false positive rate: 2.2%) for geofencing, respectively. Increasing the threshold increases the true positive rate but also increases the miss rate, as a greater number of possible cloaked items is extracted. Conversely, lowering the threshold results in fewer misses but more false positives, thus creating a trade-off. Since the purpose of this measurement study was to clarify the actual situation of the cloaking method, the thresholds were selected so that false positives could be tolerated to some extent and yet the number of missed cases would be reduced. In the following sections, we will use the above threshold values in our analysis.

IV. MEASUREMENT AND ANALYSIS

In this section, we describe the results of continuous observation of malicious hosts from multiple locations and analysis of the observation results to determine the effectiveness of Stargazer.

A. OBSERVATION DATA

As a preliminary step in observation and analysis, we collected observation targets. Most malicious hosts are relatively short-lived [27], i.e., the time from when a host is activated as

a malicious host to when it becomes inaccessible or changes to a benign host (survival time) is short. Thus, it is desirable to add malicious hosts to the observation targets as early as possible. Therefore, we collected observation targets from various information sources by selecting those that are relatively fast. Specifically, we collected 18,397 malicious URLs from malicious URL sharing sites such as URLhaus[3] and social media sites such as Twitter[4] in which many security analysts/vendors share indicators of compromise. For the Twitter information, suspicious URLs shared by prominent analysts were periodically collected. We also collected candidates of observation targets by using hash tags (e.g., *#IOC* and *#malware*). We then selected observation targets from the candidates, refanged them (e.g., converted *hxxp://example[.]com* into *http://example.com*), and registered them to the observation server. Observations were conducted once a day from November 2019 to February 2022 for the aforementioned targets, and 30,359,410 observations were obtained. This means that in this experiment, one day (24 hours) is definitively used as the length of the time series in formula 1. Note that not all the hosts were observed from the beginning of the period because the targets were added as needed during the operation.

The analysis system was applied to the aforementioned observations, and time-series and geofencing analyses were conducted. Stargazer was designed to improve the observability of malicious hosts by continuous observations from multiple geolocations. We quantitatively evaluated their observability by referring to the observation results.

Note that the results of observations with a high probability of sinkholes were excluded. The determination of whether or not a sinkhole is in fact a sinkhole was performed using observation data and DNS information inspired by [28] and [29]. In the flow using observed data, a sinkhole list of content hash values, domain names, and A/AAAA records is created and compared with the list to determine whether or not a sinkhole is present. The list was constructed and updated by adding sinkholes discovered during the operation of Stargazer. DNS information was periodically obtained, and when NS records, CNAME records, and TXT records contained keywords such as *sinkhole*, they were judged to be sinkholes. NS records define authoritative name servers, and sinkholed domain name servers may contain the keyword *sinkhole*. CNAME record defines a canonical name for a domain, and the sinkholed domain is sometimes mapped to domains that contain the keyword *sinkhole*. TXT record registers an arbitrary string, and the TXT record of the sinkholed domain sometimes indicates the domain has been sinkholed. Therefore, by referring to these records and checking whether they contain strings such as *sinkhole*, we can determine whether the observation target is sinkholed.

In the following sections, we describe the results of each analysis.

[3]https://urlhaus.abuse.ch/
[4]https://twitter.com/

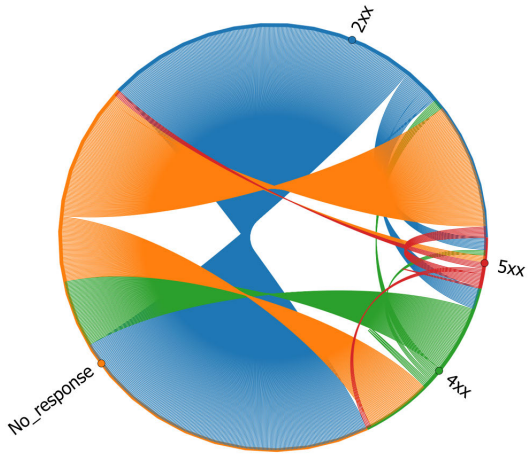


FIGURE 3. Time-series change of status code.

TABLE 2. Types of time-based cloaking and number of observations for each technique.

Cloaking type	No. of hosts	
Revive from no response	1,860	(10.11%)
Change the status code from not 2xx to 2xx	2,419	(13.14%)
Change the content	2,535	(13.78%)
Total (Unique)	3,128	(17.00%)

B. TIME-SERIES ANALYSIS

In this section, we describe the changes in the time series of each malicious host observed using Stargazer.

As shown in Table 2, changes that appear to be due to cloaking were observed in 3,128 hosts in the time series. The changes in time were identified to be divided into three main types. Specifically, the number of cases of revival from no response (1,860), change of status code from non-2xx to 2xx (2,419), and change of content (2,535). Note that the total number of each does not equal the number of unique cloaking sites (3,128), since a single host may use multiple methods.

We also checked the time-series changes in more detail from the viewpoint of status code and content. The transitions of the status codes in a time series are shown in Fig. 3 and those of the content are shown in Fig. 4, where the state before and after the transition are shown. More detailed transition figures are available in the Appendix. The content is rounded for the same type. For example, DOS-MZ, PE, and ELF are included as executable files in this observation, and all of them are listed together as executable in the figure. The breakdown of each item is shown in Table 3. The status code is also rounded for the same type. For example, *400*, *401*, *403*, etc. are listed together as *4xx* in the figure. The most common status code was the transition from *200 OK* to the no-response state. This is thought to occur when the host is abandoned as the attack ends. However, there were cases in which the host revived from the unresponsive state. Many of

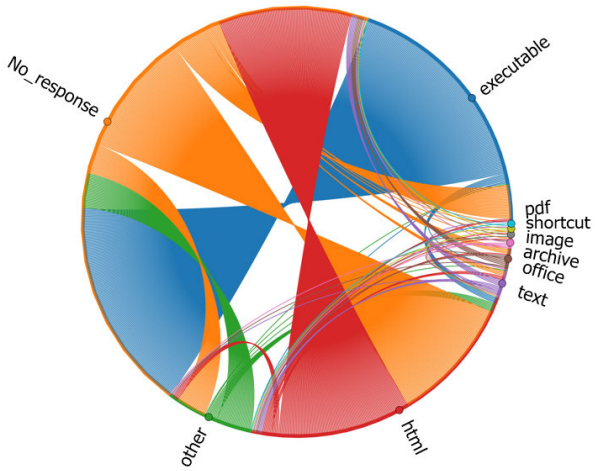


FIGURE 4. Time-series change of content.

TABLE 3. Components of each category.

Content category	Components
executable	DOS-MZ, PE, ELF, shell script
html	HTML
image	JPEG, PNG
office	Excel, Word
archive	Zip, ACE, JAR
text	Text
pdf	PDF
shortcut	MS Windows shortcut
other	empty, data, very short file (no magic)

the transitions within the intermediate states go back and forth between *200 OK*, *404 Not Found*, and no-response. A certain number of status code changes to *2xx*, which Stargazer detects as a sign of the start of an attack, also existed, suggesting that it is necessary to consider the possibility of re-activation even once there is no response. Next, as with the status codes, the most common transition to the unresponsive state was for the contents. This is also thought to be caused by the end of the attack when the host is abandoned. Regarding other changes, the change to HTML was frequently observed. In other cases, as a type of observation evasion, executable files were distributed once and replaced with harmless HTML after the second time. The detection of these more dangerous changes should improve the response speed and resolution against attacks.

To summarize, several time-based cloaking methods are observed and we defined three categorize of them. In addition, continuous observation makes it possible to observe even hosts that show highly suspicious characteristics only temporarily and with changes. We have demonstrated that it is possible to detect the re-activation of malicious hosts by using changes in status codes and content, together with case studies.

TABLE 4. Types of geofencing and number of observations for each technique.

Cloaking type	No. of hosts	
Change the hash value of the content	2,708	(14.71%)
Change the format of the content	1,467	(7.97%)
Change the format of the status code	1,553	(8.44%)
Respond only to specific geolocations	1,102	(5.99%)
Total (Unique)	2,716	(14.76%)

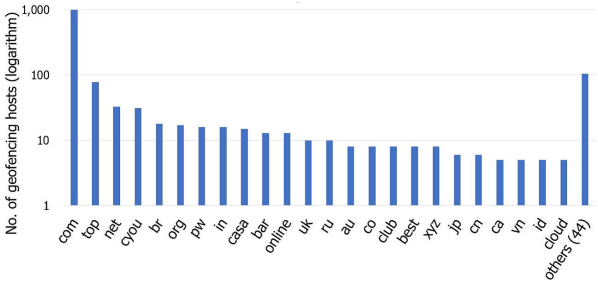


FIGURE 5. Number of geofencing hosts' TLDs.

TABLE 5. Number of target geolocations.

No. of targets	No. of hosts	
1	2,281	(83.98%)
2	222	(8.17%)
3	93	(3.43%)
More than 4	120	(4.42%)
Total	2,716	(100.00%)

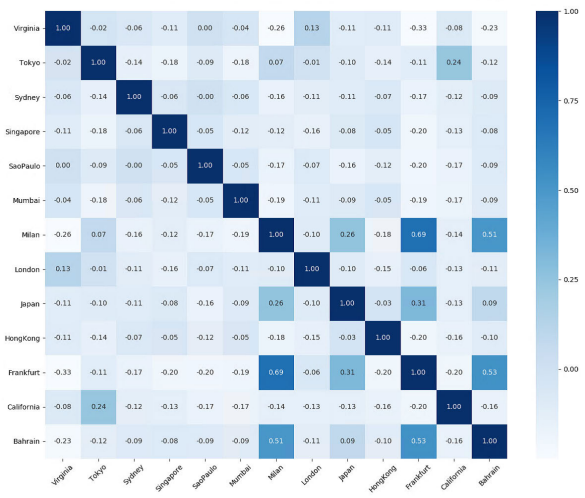


FIGURE 6. Correlation coefficients for each geolocation.

C. GEOFENCING ANALYSIS

In this section, we describe the hosts with geofencing detected using Stargazer.

The geofencing observed in this study is shown in Table 4. In this observation, we observed four major evasion methods related to geofencing. The most common method was to return contents with different hash values depending on the geolocation, with 2,708 cases. We found 1,467 cases of changing the content format in accordance with the geolocation. For example, this method returns executable content only to certain geolocations and harmless content, such as simple HTML files, to other geolocations. There were 1,553

hosts that returned a *200 OK* status code only to a specific geolocation and 1,102 hosts that responded only to a specific geolocation and not to other geolocations. Since a single host may have multiple methods, e.g., changing both content and status codes depending on the geolocation, the total number of all methods shown in Table 4 does not equal the unique number of hosts with regional characteristics, which is 2,716.

Fig. 5 shows the ratio of top-level domains (TLDs) to geofenced domains. Seventy-four TLDs were identified, with *.com* being the most common (1,001) and *.net* the third most common (33), indicating that many well-known TLDs were observed. However, relatively new TLDs, such as *.cyou* and *.club*, were also found in the top rankings. Since domain names in these TLDs are relatively cheaper to acquire than those in well-known TLDs, it is assumed that adversaries are using them as disposable domain names.

The number of geolocations targeted by the cloaking hosts is shown in Table 5. Most, 83.98%, targeted a single geolocation, but there were those that targeted two or more geolocations. The Japan sensor was the only one deployed in a PF that is not AWS, which may have something to do with this observation. For example, since the IP address range of AWS is public,[5] it is possible that cloaking is implemented to prevent malicious content from being returned for accesses from this IP address. The correlation coefficients for each geolocation are shown in Fig. 6. As mentioned above, most are single targets, but there is a loose correlation between Frankfurt, Milan, and Bahrain. We checked the hosts that simultaneously targeted these and found that many were reported as belonging to the TA551/Shathak group, which targeted at least German and Italian speakers [30], [31]. In addition, although we have not been able to find any information on this, it is possible that Bahrain may also have been targeted in the same attack campaign.

In summary, by conducting observations from multiple observation sites, it is possible to observe even hosts with regional characteristics. We compared the observation results among the observation sensors and clarified the analysis-avoidance method.

D. OBSERVABILITY

In this research, we verified the observability of malicious contents. Specifically, the hash values of the observed contents were used to query VirusTotal[6] to verify their existence. In this experiment, 2,208 contents were randomly selected and counted on two perspectives: whether they were present in VirusTotal and whether they had been cloaked. Note that HTML contents were excluded in this experiment because most of them are benign.

The verification results are shown in Table 6. First, 1,224 contents (81.06%) of non-regional contents were found in VirusTotal. This is because the contents are not cloaked and can be accessed from any geolocation or in any time; as a result, a high percentage of the contents were submitted to VirusTotal. However, 60.32% of the cloaked contents were not found in VirusTotal. Thus, contents distributed by cloaking hosts had a lower presence rate in VirusTotal than those distributed by non-cloaking hosts. The null hypothesis that there is no association between the presence or absence of cloning and the presence or absence of VirusTotal was rejected by the chi-square test at the 0.05 level of significance ($p \fallingdotseq 3.22E\text{-}83 < 0.05$). From these point of view, we can say that Stargazer improves observability.

As described above, we have shown that the observability of malicious hosts can be improved by continuously observing them from multiple geolocations.

TABLE 6. Presence/absence of geofenced contents in VirusTotal.

	Presence		Absence		Total	
Geofenced	277	(39.68%)	421	(60.32%)	698	(100%)
Not geofenced	1,224	(81.06%)	286	(18.94%)	1,510	(100%)
Total					2,208	-

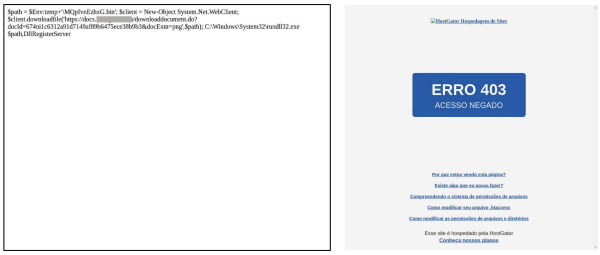


FIGURE 7. Examples of cloaked malicious host. The left one is accessed from Milan and Bahrain and right one is accessed from other locations.

V. CASE STUDY

In this section, we discuss case studies related to time-series changes of malicious hosts and cloaking. Typical examples include observation evasion, such as the transition between active and dormant states described above, and observation evasion by geofencing using methods such as those listed in Table 4. Our case studies explore some of the more characteristic examples.

A. CASE 1: TIMING-BASED CLOAKING

In the first case, the malicious host was taken down and then restored. This site was reported around November 19, 2020 as a site distributing *Dridex* executables disguised as pdf extensions. The site suspended malware distribution on November 20, 2020, went down for about three weeks with HTTP status code *503*, and started distributing the same malware again on December 11, 2020. Finally, it became *404 Not Found* on or around March 25, 2021, and was not activated until the domain was destroyed. However, it became *503* once immediately after the report, and was re-activated after an interval, meaning that it survived for about four months in effect. This is one of the longest-lived attacking hosts, which are typically said to be relatively short-lived. From the attacker's perspective, the simple operation of reactivating the site after it has been down for several weeks prolongs the life of the

[5]http://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html
[6]https://www.virustotal.com/

attack site associated with a single domain and improves the return on investment against attacks.

B. CASE 2: GEOFENCING

The second case is a site that provides clients accessing from Milan and Bahrain with a script that downloads a secondary sample (*Ursnif*) (Fig. 7, left), and returns a *403 Forbidden* (Fig. 7, right) for clients from other locations. Several hosts with similar characteristics were observed at the same time. Some malicious hosts with similar characteristics returned *404 Not Found* instead of *403 Forbidden*, and some returned empty files with *200 OK* unless the client was the attack target. Some of them did not return any response, throwing errors such as *'Connection aborted.', RemoteDisconnected('Remote end closed connection without response')*, except for the attack target. Some returned malicious responses, such as executable files only for the first access from the attack target, and returned harmless responses for the second and subsequent accesses, just like accesses from other non-attack areas. In all cases, the attackers downloaded Dynamic Linking Libraries (DLLs) placed on file-sharing services with randomized file names.

As for the group of sites that seemed to be related to the above-mentioned campaign, both the malware download site and the C2 server were cloaked. However, while the malware download site had a survival time of about one week, the C2 server had a survival time of more than two weeks and was shared as the C2 server for each malware. It is assumed that the first malware distribution site, which is relatively easy to detect due to its accessibility, has a short survival period, while the C2 server in the latter stage is used by multiple samples with a long survival period, thereby reducing the cost of the attack.

C. CASE 3: MULTIPLE TIMING-BASED CLOAKING WITH AWS DETECTION

The third case was a combination of multiple time-based cloaking and AWS detection. We present two similar examples identified in different paths under the same domain. Fig. 8 shows the status code, IP address, content type, and hash value (SHA256/ssdeep) of the executable file for each observation timing. These were all reported around May 20, 2021 as distribution hosts for *RedLine Stealer*, and Stargazer began observations on June 16, 2021. First, as shown in the figure, the IP address changes in a short span, and the status code (*200 OK, 404 Not Found, 503 Service Temporarily Unavailable*) changes depending on the access destination according to the IP address, and even when the IP address is *200 OK*, it is different. We confirmed that different executable files come down from the server, which suggests the attacker manipulates files on the server at any time. Also, different executables are downloaded at different times and with different status codes, even for the same IP address. All of these samples were packed, and a simple calculation of the similarity between each sample using *ssdeep* showed a result of 0. However, when we checked each sample's behavior report in *JoeSandbox*,[7] we found that for those that existed, they were all determined to be *RedLine* and connected to the same C2 server. Therefore, it can be said that they are basically equivalent or similar samples.

For this case, we checked the first submission of each executable file in VirusTotal and found that most of them were later than the date and time of the first observation. We assume that the intention was to evade hash value-based detection by constantly distributing new specimens that were not in the VirusTotal at the time of the observation. In addition, at 34[.]75[.]49[.]***, we observed a behavior that would lead to the detection of AWS Specifically, *503 Service Temporarily Unavailable* was returned when accessed from AWS, whereas an executable file was returned with *200 OK* when accessed from the on-premises environment. Since no AWS detection behavior was observed at other times on the same host, we confirmed that AWS can be detected and, in some cases, the behavior can be changed accordingly, although there is a possibility of unintentional discrimination due to misconfiguration, etc.

In addition, screenshots taken with a headless browser showed that only html with *404 Not Found* was displayed, regardless of the timing of access or the actual status code. This was presumably done to avoid detection by making it look like a *404* when accessed from a browser. This host was in operation until July 11, 2021, which means it had survived about two months from the date of the report.

This example demonstrates an attempt to evade detection by using a variety of techniques, including content, status code, AWS detection, and browser detection, in addition to a technique known as fast-fluxing [32], which involves briefly reassigning IP addresses. As a result, it survived about two months, providing an increased return on investment for the attacker.

The case studies presented in this section were chosen to represent the more common characteristics of each of the cloaking techniques described in the previous sections. Although Stargazer was able to observe some of the species in this study, they were all likely to slip past detection by simple observation, and in fact most of them had survived for a relatively long period of time. This highlights the importance of utilizing Stargazer for early detection of such items as well.

VI. DISCUSSION

A. OBSERVABILITY

In Stargazer, cloaking in accordance with the access source is made difficult by installing observation sensors in multiple geolocations. However, another cloaking method involves collecting the IP addresses of researchers, creating a reject list, and returning harmless content to accesses from the IP addresses in the list [33]. We actually found a host, which denied accessing from AWS's IP address range, as shown in the case study. There is a method of returning malicious content only to the IP address of a malware-infected

[7] https://www.joesandbox.com/

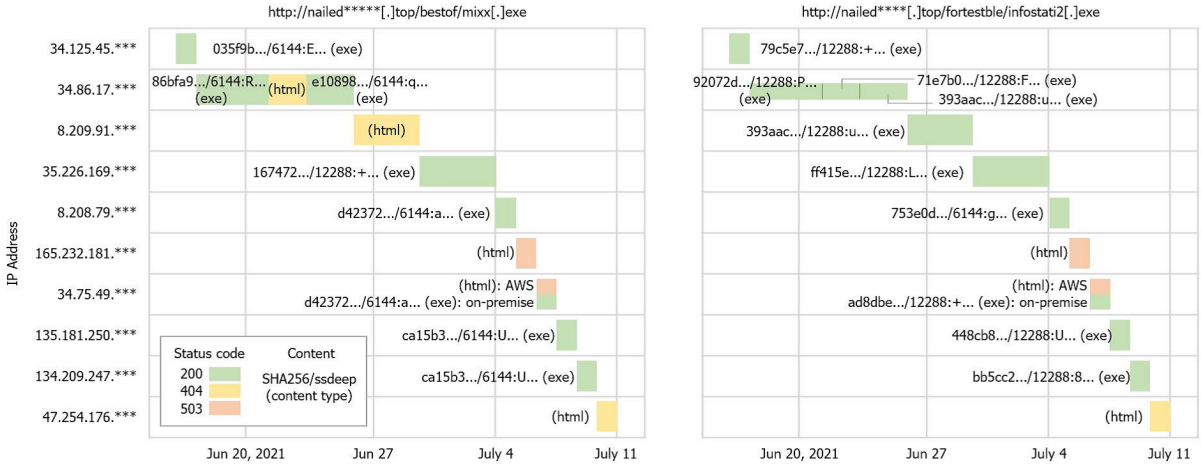


FIGURE 8. Time-series changes of a domain name about assigned IP address and its HTTP status code with AWS detection. Note that a part of URLs and IP addresses are anonymized.

terminal [34]. However, Stargazer can make this avoidance more difficult by changing the IP address of the observation sensor periodically or by placing the sensor in a location other than the AWS (i.e., in another Infrastructure as a Service (IaaS) or Virtual Private Network (VPN)). The Onion Router (Tor) could also be used. By utilizing different types of observation sensors, we should be able to improve observability by making it more difficult to avoid observation by cloaking, and to estimate the target of cloaking and extract more suspicious sites.

Moreover, limitations on the monitoring resources make it difficult to perform monitoring indefinitely. Although we conducted observations over an even span of time, we expect to maximize scalability by focusing on hosts that are considered to have higher maliciousness (e.g., hosts with time-series changes or regional characteristics) and reducing the frequency of observation of hosts that are not. In addition, while some malicious hosts may change their behavior in a short period of time, others may not change their behavior at all for a long period of time. Thus, it is desirable to somehow determine the time-series length for each malicious host in the future.

Furthermore, the observation results are subject to selection bias. For example, many executable files were observed in this experiment, but this was partly due to the fact that URLhaus, where the information was collected, had a high concentration of executable file postings during the collection period. Although it is inherently difficult to solve this problem, we are trying to mitigate it by using Twitter and other information sources in addition to URLhaus. Also, the selection bias should be alleviated as we continue to collect and add observation targets.

B. FALSE POSITIVE

Stargazer can potentially be fraught with false positives. For example, in time-series analysis, a dormant site reactivated as a harmless site may be erroneously determined as dangerous. In the same way, if the file-sharing function of a rental server or a benign site is temporarily abused, it may be over-detected as a change or activation because each site returns *404 Not Found*, etc. after being destroyed. However, such false positives can be suppressed by adding those benign sites to the allow list.

In this study, the monitoring targets were set based on information from general users on sites such as URLhaus and Twitter, so there is a possibility that the information is incorrect or that incorrect information is intentionally registered. However, both URLhaus and Twitter require user registration and are thus more reliable than anonymous information. Another potential limitation is that registering benign hosts for monitoring based on incorrect information would overwhelm Stargazer's monitoring resources. However, this problem can be mitigated by utilizing allow lists to exclude benign hosts.

C. RESEARCH ETHICS

In the observation of Stargazer, we use HTTP GET, ping, etc., which can occur in normal use, and we do not use malicious communication. Due to the fact that we try to access malicious hosts, we may receive alerts and requests for various responses from the sinkhole administrators and IaaS providers. We organized a team to respond to these communications as we conducted the observation. We received three inquiries during the observation period and responded to all of them within 24 hours.

VII. RELATED WORK

There has been extensive research on cloaking detection, particularly on the cloaking of phishing sites and fake AV distribution sites, as they have conflicting requirements: on the one hand, they want to appear at the top of search engines by Search Engine Optimization in order to reach more attack targets, and on the other hand, they do not want to be analyzed by researchers. CrawlPhish [15] has detected

cloaking in phishing and conducted a large-scale analysis. Invernizzi et al. [16] proposed a cloaking detection method and conducted a large-scale survey and analysis of cloaking related to URLs associated with search and advertising. Wu et al. proposed a method for detecting semantic cloaking pages by using content differences retrieved by web crawlers and web browsers [24]. Mansoori et al. [35] attempted to detect geofencing by concurrently accessing malicious hosts from clients located in six geolocations and revealed the correlation between TLDs and targeted locations. Drury et al. analyzed the life cycle of phishing sites to determine their survival time and campaign-specific characteristics [14]. Bijmans et al. identified attacks using off-the-shelf phishing kits, observed them, and derived TTPs [17]. However, most studies are tailored to use features specific to hosts with web pages (including cloaking sites), such as screenshot differences and JavaScript structures. In this study, we did not limit ourselves to specific types of malicious hosts but rather presented a general method that uses the differences of long-term observations from multiple geolocations. There is some studies that observes cloaking without limiting it to specific types of attacks. However, continuous observation and analysis of the cloaking content was not conducted. In this study, we improved the observability by continuous observation using observation sensors with wider coverage.

Much research has been conducted on the observation of suspicious hosts. CyberProbe [36] uses active scanning to detect C2 servers and bots in the Listen state, while, Soska et al.'s method uses observation data to predict whether a site will be compromised [37]. EvilSeed [38] collects malicious hosts by generating efficient patrol queries used by client-based honeypots on the basis of known malicious host information. Although these methods are useful, they do not target continuous observation. In contrast, Stargazer shows that it is possible to detect changes in time series and to improve observability by conducting continuous observations.

Many other studies have focused their observations on malware distribution. Rossow et al. [39] observes malware downloaders over a long period of time and discusses the possibility that the source IP addresses are blocked by attackers during long-term observations. Jaun et al. [18] observed malware distributed via PPI and determined that some malware families are biased in the regions where they are distributed. Thomas et al. [19] similarly observed malware distributed via PPI. Inspired by these studies, we conducted systematic measurement study focusing on cloaking malware distribution hosts and C2 servers.

TARDIS [40] is a method for detecting attacks targeting Contents Management Systems (CMSs). It uses content continuously retrieved from websites to detect attacks. Barron et al. operated honeypots deployed in multiple geolocations and analyzed attacks from the perspective of honeypot locations [41]. Another study using honeypots [42] showed that the characteristics of attacks vary depending on the region. The results of Spoki [43], a method for observing Internet

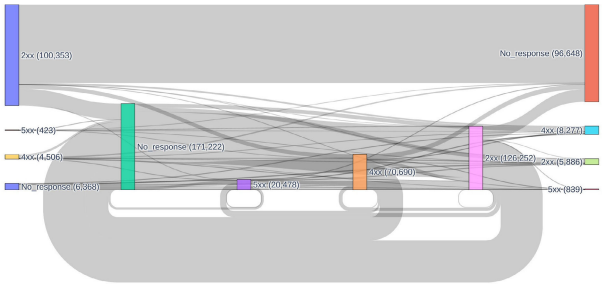


FIGURE 9. Time-series change of status code (sankey diagram).

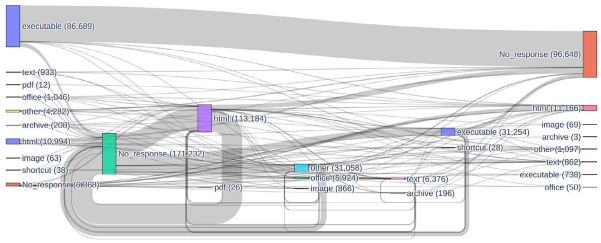


FIGURE 10. Time-series change of content (sankey diagram).

scanning activity, showed that different regions have different scanning activity characteristics. Augur [44] detects the beginning and end of censorship by continuously observing websites from multiple geolocations. ICLab [45] and Censored Planet [46] are also censorship-detection systems with sensors installed in multiple geolocations. These systems are similar to Stargazer in that they execute observations continuously and from multiple locations, but they have different objectives. Stargazer detects changes in the time series in a manner robust against cloaking by using the observation results from multiple locations, thus demonstrating improvement in observability against malicious hosts.

VIII. CONCLUSION

In this study, we observed a total of 18,397 malicious hosts for over two years using Stargazer, which actively observes malicious hosts Internet-wide and detects geofencing and time-based cloaking, to elucidate its actual characteristics. Our observations confirmed that cloaking techniques are omnipresent among malicious hosts. These include malicious hosts that evade detection through time-based cloaking, geofencing, and a combination of the two, which are difficult to detect without long-term, multiregional observations, which Stargazer can provide.

In addition, we confirmed that malicious hosts performing cloaking include those that survive for relatively long periods of time and those whose contents are not present in VirusTotal. These results clarify the mechanisms underlying cloaking technology and suggest that there are cloaking sites that are not easy to observe with existing technology. We believe our findings will be helpful for the design of future observation systems and contribute to the development of cloaking detection methods.

ACKNOWLEDGMENT

The authors would like to express their deepest gratitude to all those involved for their useful advice and cooperation in conducting this research.

APPENDIX TIME-SERIES CHANGES

This appendix further details the time-series changes of the contents and status codes discussed in Section IV-B. The detailed transitions of the status codes in a time series are shown in Fig. 9, and the transitions of the content are shown in Fig. 10 as a Sankey diagram. In each figure, the left end is the first observation, the center is the intermediate state, and the right end is the final observation. The transitions from the intermediate state to another intermediate state are also included. As in Section IV-B, we can confirm here that many of the transitions of status codes within the intermediate states go back and forth between *200 OK*, *404 Not Found*, and no-response.

REFERENCES

[1] *Unit42: Case Study: Emotet Thread Hijacking, an Email Attack Technique*. Accessed: Dec. 1, 2022. [Online]. Available: https://unit42.paloaltonetworks.com/emotet-thread-hijacking/

[2] *JPCERT/CC: Malware Used by Lazarus After Network Intrusion*. Accessed: Dec. 1, 2022. [Online]. Available: https://blogs.jpcert.or.jp/en/2020/08/Lazarus-malware.html

[3] L. Invernizzi, S. Miskovic, R. Torres, S. Saha, S.-J. Lee, M. Mellia, C. Kruegel, and G. Vigna, "NAZCA: Detecting malware distribution in large-scale networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014.

[4] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, and T. Dumitraş, "The dropper effect: Insights into malware distribution with downloader graph analytics," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1118–1129.

[5] A. Oest, Y. Safei, A. Doupé, G. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, May 2018, pp. 1–12.

[6] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, and A. Doupé, "PhishTime: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 379–396.

[7] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G.-J. Ahn, "Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 361–377.

[8] E. De Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Z. Shafiq, "Paying for likes?: Understanding Facebook like fraud using honeypots," in *Proc. Conf. Internet Meas. Conf.*, Nov. 2014, pp. 129–136.

[9] S. Farooqi, G. Jourjon, M. Ikram, M. A. Kaafar, E. De Cristofaro, Z. Shafiq, A. Friedman, and F. Zaffar, "Characterizing key stakeholders in an online black-hat marketplace," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, Apr. 2017, pp. 17–27.

[10] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna, "The underground economy of spam: A botmaster's perspective of coordinating large-scale spam campaigns," in *Proc. 4th USENIX Workshop Large-Scale Exploits Emergent Threats*, 2011.

[11] M. Cova, C. Leita, O. Thonnard, A. D. Keromytis, and M. Dacier, "An analysis of rogue AV campaigns," in *Recent Advances in Intrusion Detection*. Berlin, Germany: Springer, 2010, pp. 442–463.

[12] T. Koide, D. Chiba, M. Akiyama, K. Yoshioka, and T. Matsumoto, "It never rains but it pours: Analyzing and detecting fake removal information advertisement sites," in *Detection of Intrusions and Malware, and Vulnerability Assessment*. Cham, Switzerland: Springer, 2020, pp. 171–191.

[13] B. Z. H. Zhao, M. Ikram, H. J. Asghar, M. A. Kaafar, A. Chaabane, and K. Thilakarathna, "A decade of mal-activity reporting: A retrospective analysis of internet malicious activity blacklists," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Jul. 2019, pp. 193–205.

[14] V. Drury, L. Lux, and U. Meyer, "Dating phish: An analysis of the life cycles of phishing attacks and campaigns," in *Proc. 17th Int. Conf. Availability, Rel. Secur.*, Aug. 2022, pp. 1–11.

[15] P. Zhang, A. Oest, H. Cho, Z. Sun, R. Johnson, B. Wardman, S. Sarker, A. Kapravelos, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupé, and G. Ahn, "CrawlPhish: Large-scale analysis of client-side cloaking techniques in phishing," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 1109–1124.

[16] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J. Picod, and E. Bursztein, "Cloak of visibility: Detecting when machines browse a different web," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 743–758.

[17] H. Bijmans, T. Booij, A. Schwedersky, A. Nedgabat, and R. van Wegberg, "Catching phishers by their bait: Investigating the Dutch phishing landscape through phishing kit detection," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 3757–3774.

[18] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring pay-per-install: The commoditization of malware distribution," in *Proc. 20th USENIX Secur. Symp.*, 2011, pp. 1–13.

[19] K. Thomas, J. A. E. Crespo, R. Rasti, J.-M. Picod, C. Phillips, M.-A. Decoste, C. Sharp, F. Tirelo, A. Tofigh, M.-A. Courteau, L. Ballard, R. Shield, N. Jagpal, M. A. Rajab, P. Mavrommatis, N. Provos, E. Bursztein, and D. McCoy, "Investigating commercial pay-per-install and the distribution of unwanted software," in *Proc. 25th USENIX Conf. Secur. Symp.*, 2016, pp. 721–738.

[20] *Check Point Research: Google is on Guard: Sharks Shall not Pass!* Accessed: Dec. 1, 2022. [Online]. Available: https://research.checkpoint.com/2022/google-is-on-guard-sharks-shall-not-pass/

[21] *Gamaredon APT Targets Ukrainian Government Agencies in New Campaign*. Accessed: Dec. 1, 2022. [Online]. Available: https://blog.talosintelligence.com/2022/09/gamaredon-apt-targets-ukrainian-agencies.html

[22] *FOREGENIX: An Overview on Purple Fox*. Accessed: Dec. 1, 2022. [Online]. Available: https://www.foregenix.com/blog/an-overview-on-purple-fox

[23] S. Fujii, T. Sato, S. Aoki, Y. Tsuda, Y. Okano, T. Shigemoto, N. Kawaguchi, and M. Terada, "Continuous and multiregional monitoring of malicious hosts," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 2101–2103.

[24] B. Wu and B. D. Davison, "Detecting semantic cloaking on the web," in *Proc. 15th Int. Conf. World Wide Web*, May 2006, pp. 819–828.

[25] N. Samarasinghe and M. Mannan, "On cloaking behaviors of malicious websites," *Comput. Secur.*, vol. 101, Feb. 2021, Art. no. 102114.

[26] *Unit42: Detecting Patient Zero Web Threats in Real Time With Advanced URL Filtering*. Accessed: Dec. 1, 2022. [Online]. Available: https://unit42.paloaltonetworks.com/patient-zero-web-threats/

[27] M. Akiyama, M. Iwamura, Y. Kawakoya, K. Aoki, and M. Itoh, "Design and implementation of high interaction client honeypot for drive-by-download attacks," *IEICE Trans. Commun.*, vol. 93, no. 5, pp. 1131–1139, 2010.

[28] E. Alowaisheq, P. Wang, S. Alrwais, X. Liao, X. Wang, T. Alowaisheq, X. Mi, S. Tang, and B. Liu, "Cracking the wall of confinement: Understanding and analyzing malicious domain take-downs," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019.

[29] *McAFee: Rovnix Downloader Updated With SinkHole and Time Checks*. Accessed: Dec. 1, 2022. [Online]. Available: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/rovnix-downloader-sinkhole-time-checks/

[30] *MITRE ATT&CK: TA551, GOLD CABIN, Shathak, Group G0127*. Accessed: Dec. 1, 2022. [Online]. Available: https://attack.mitre.org/groups/G0127/

[31] *Unit42: TA551: Email Attack Campaign Switches From Valak to IcedID*. Accessed: Dec. 1, 2022. [Online]. Available: https://unit42.paloaltonetworks.com/ta551-shathak-icedid/

[32] Z. Guo and Y. Guan, "Active probing-based schemes and data analytics for investigating malicious fast-flux web-cloaking based domains," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–9.

[33] K. Zeeuwen, M. Ripeanu, and K. Beznosov, "Improving malicious URL re-evaluation scheduling through an empirical study of malware download centers," in *Proc. Joint WICOW/AIRWeb Workshop Web Qual.*, Mar. 2011, pp. 42–49.

[34] M. Mansoori, I. Welch, K.-K.-R. Choo, R. A. Maxion, and S. E. Hashemi, "Real-world IP and network tracking measurement study of malicious websites with HAZOP," *Int. J. Comput. Appl.*, vol. 39, no. 2, pp. 106–121, Apr. 2017.

[35] M. Mansoori and I. Welch, "Geolocation tracking and cloaking of malicious web sites," in *Proc. IEEE 44th Conf. Local Comput. Netw. (LCN)*, Oct. 2019, pp. 274–281.
[36] A. Nappa, Z. Xu, M. Z. Rafique, J. Caballero, and G. Gu, "CyberProbe: Towards internet-scale active detection of malicious servers," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014.
[37] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 625–640.
[38] L. Invernizzi and P. M. Comparetti, "EvilSeed: A guided approach to finding malicious web pages," in *Proc. IEEE Symp. Secur. Privacy*, May 2012, pp. 428–442.
[39] C. Rossow, C. Dietrich, and H. Bos, "Large-scale analysis of malware downloaders," in *Proc. 9th Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*, 2012, pp. 42–61.
[40] R. Pai Kasturi, Y. Sun, R. Duan, O. Alrawi, E. Asdar, V. Zhu, Y. Kwon, and B. Saltaformaggio, "TARDIS: Rolling back the clock on CMS-targeting cyber attacks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 1156–1171.
[41] T. Barron and N. Nikiforakis, "Picky attackers: Quantifying the role of system properties on intruder behavior," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, Dec. 2017, pp. 387–398.
[42] M. Wählisch, S. Trapp, C. Keil, J. Schönfelder, T. C. Schmidt, and J. Schiller, "First insights from a mobile honeypot," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 305–306, Sep. 2012.
[43] R. Hiesgen, M. Nawrocki, A. King, A. Dainotti, C. T. Schmidt, and M. Wählisch, "Spoki: Unveiling a new wave of scanners through a reactive network telescope," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 431–448.
[44] P. Pearce, R. Ensafi, F. Li, N. Feamster, and V. Paxson, "Augur: Internet-wide detection of connectivity disruptions," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 427–443.
[45] A. A. Niaki, S. Cho, Z. Weinberg, N. P. Hoang, A. Razaghpanah, N. Christin, and P. Gill, "ICLab: A global, longitudinal internet censorship measurement platform," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 135–151.
[46] R. Sundara Raman, P. Shenoy, K. Kohls, and R. Ensafi, "Censored planet: An internet-wide, longitudinal censorship observatory," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 49–66.

SHOTA FUJII received the M.E. and Ph.D. degrees from Okayama University, Japan, in 2016 and 2023, respectively. He has been with Hitachi Ltd., since 2016. His research interests include computer security and virtualization technology. He is a member of IPSJ.

TAKAYUKI SATO has been with Hitachi Ltd., since 2003. He is engaged in business planning for cyber security related services, internal security strategy, and security design/operation of customer systems.

SHO AOKI has been with Hitachi Ltd., since 2013. He is currently a member of the Hitachi Incident Response Team (HIRT), where he works on vulnerability countermeasures and incident response.

YU TSUDA received the Ph.D. degrees from Kyoto University, Kyoto, Japan, in 2013 and 2016, respectively.

From 2013 to 2018, he was a Researcher with the National Institute of Information and Communications Technology (NICT). Since 2018, he has been a Senior Researcher with NICT. His research interests include countermeasures against targeted attacks and cyber threat intelligence. He is also interested in Capture the Flag (CTF) games. He is currently a member of the SECCON Executive Committee, which is the largest CTF organizer in Japan.

NOBUTAKA KAWAGUCHI received the M.E. and Ph.D. degrees from Keio University, Japan, in 2005 and 2008, respectively.

He has been with Hitachi Ltd., since 2008. His research interests include network security and malware detection. He is a member of ACM and IPSJ.

TOMOHIRO SHIGEMOTO received the M.E. degree from Osaka University, Japan, in 2006, and the Ph.D. degree from Meiji University, Japan, in 2019. He has been with Hitachi Ltd., since 2006. His research interests include network security and malware detection. He is a member of IPSJ.

MASATO TERADA received the M.E. degree from Chiba University, Japan, in 1986, and the Ph.D. degree from Keio University, Japan, in 2006.

Since joining Hitachi Ltd., in 1986, he has been engaged in research on cyber security with the Research and Development Group and the Hitachi Incident Response Team (HIRT). Since 2019, he has been a Professor with Tokyo Denki University. He is currently an Expert Committee of the JPCERT/CC, a Researcher of the Information-Technology Promotion Agency (IPA) Security Center, a Guest Assistant Professor with Chuo University, a Steering Committee Chair with the Nippon CSIRT Association (NCA), and a Steering Committee of the ICT-ISAC, Japan.