2019 IEEE Symposium on Security and Privacy

Resident Evil: Understanding Residential IP Proxy as a Dark Service

Xianghang Mi*, Xuan Feng*, Xiaojing Liao*, Baojun Liu†,
XiaoFeng Wang*, Feng Qian*, Zhou Li‡, Sumayah Alrwais§, Limin Sun¶, Ying Liu†
*Indiana University Bloomington, †Tsinghua University, ‡IEEE Member,
§King Saud University, ¶Institute of Information Engineering, CAS
*{xmi, xf1, xliao, xw7, fengqian}@indiana.edu, †lbj15@mails.tsinghua.edu.cn,
liuying@cernet.edu.cn, ‡lzcarl@gmail.com, §salrwais@ksu.edu.sa, ¶sunlimin@iie.ac.cn,

***Abstract*—An emerging Internet business is residential proxy (RESIP) as a service, in which a provider utilizes the hosts within residential networks (in contrast to those running in a datacenter) to relay their customers' traffic, in an attempt to avoid server-side blocking and detection. With the prominent roles the services could play in the underground business world, little has been done to understand whether they are indeed involved in Cybercrimes and how they operate, due to the challenges in identifying their RESIPs, not to mention any in-depth analysis on them.**

In this paper, we report the *first* study on RESIPs, which sheds light on the behaviors and the ecosystem of these elusive gray services. Our research employed an infiltration framework, including our clients for RESIP services and the servers they visited, to detect 6 million RESIP IPs across 230+ countries and 52K+ ISPs. The observed addresses were analyzed and the hosts behind them were further fingerprinted using a new profiling system. Our effort led to several surprising findings about the RESIP services unknown before. Surprisingly, despite the providers' claim that the proxy hosts are willingly joined, many proxies run on likely compromised hosts including IoT devices. Through cross-matching the hosts we discovered and labeled PUP (potentially unwanted programs) logs provided by a leading IT company, we uncovered various illicit operations RESIP hosts performed, including illegal promotion, Fast fluxing, phishing, malware hosting, and others. We also reverse engineered RESIP services' internal infrastructures, uncovered their potential rebranding and reselling behaviors. Our research takes the first step toward understanding this new Internet service, contributing to the effective control of their security risks.

I. Introduction

In October 2016, a spree of massive distributed denial-of-service (DDoS) attacks temporarily brought down the Domain Name System (DNS) operated by Dyn, a leading DNS provider, causing major Internet platforms and services (such as Amazon, Netflix, Paypal, Twitter et al.) to be unavailable across Europe and North America. What is remarkable about this attack is that the traffic observed was found to originate from 65,000 infected residential hosts, including home routers, web cameras, and digital video recorders [55]. Not only did these hosts jointly produce an overwhelming volume at 600 Gbps, one of the largest on record, but their residential IP addresses made the attack requests they issued less differentiable from legitimate ones, and therefore hard to detect and block by the victim.

Residential IP Proxy as a Service. Recent years have witnessed increasing demands for such *residential IPs* (those belonging to ISP's dynamically assigned IPs, particularly to home owners) as intermediaries to circumvent the restrictions imposed by target services, for the purposes such as aggressive resource access (e.g., registering multiple accounts), data scraping, and others. This emerging market gives rise to a new service model we call *Residential IP Proxy as a Service* (RPaaS), offered by companies like Luminati [3], StormProxies [49], Microleaves [38], etc. These providers all control a large number of residential hosts, which they claim joined their services willingly, to proxy their customers' communication with any Internet target. Once abused, these residential proxies can outperform conventional public proxies or even anonymity networks to help their clients masquerade as clean and benign sources to communicate with the targets. Such communication may violate the target's service terms at the very least (e.g., data scraping, blackhat Search Engine Optimization(SEO)) and is likely associated with more sinister events such as the aforementioned DDoS, due to the permissiveness of the RPaaS providers in terms of what can be done through their proxies.

With their importance to the illicit activities, residential proxy (RESIP) services, however, are still less understood. One may ask whether these services indeed use residential hosts as they claim, and if so, how they recruit these hosts, and whether they are involved in malicious activities. Also unclear are their infrastructures and ecosystems, particularly the ways they promote, operate their businesses and also work with each other. Answers to these questions are critical for determining the role these services play in Cybercrimes, which could potentially help identify an effective way to mitigate the threats we are facing today, for example, through controlling accesses to these services.

Our study. Understanding RESIP service is by no means trivial. Unlike open proxies, which can be easily found online, RESIP IPs are not publicized directly and can only be reached through the mediation of a RESIP provider. Even given a proxy's IPs, no existing techniques can tell us whether they are indeed residential, not to mention finding out whether their hosts are indeed willing participants or just controlled bots. Even more challenging is to determine whether these proxies are malicious and to understand their illicit activities, since all we can observe are just dynamic IPs shared by a set of hosts. As a result, the traffic associated with the IPs describes those

© 2019, Xianghang Mi. Under license to IEEE.
DOI 10.1109/SP.2019.00011

hosts' collective activities and it is less clear how to separate the good behaviors (when the IP is assigned to a legitimate host) from the bad ones (when it is given to a compromised host). Further without observing the internal operations of a RESIP service, understanding its infrastructure and connections with other services is difficult.

In our research, we addressed these challenges with a suite of innovative techniques, which enabled us to perform a large-scale study, first of its kind, to understand the way RESIP service is utilized for illicit purposes. Our study was based upon a novel framework for automatic discovery of RESIP IPs from related services. More specifically, we first purchased the services from commercial RESIP service providers and ran a set of clients to communicate with our web servers through these services. Traffic in the communication was carefully marked with unique sub-domains and other parameters to help the servers identify the IPs of the RESIPs, to enable our DNS system to find the DNS resolvers, and to ensure the proxied traffic of RESIPs is captured (§IV-C). The IPs found in this way were further analyzed to extract a set of unique Whois and DNS features for determining whether they are indeed residential. Further these IPs were probed by a novel, high-performance host profiling system that concurrently fingerprints the hosts behind millions of IPs, both from the clients and the servers under our control. Our fingerprinting technique ensures that the target of our analysis is always the RESIP, despite its highly fluctuating IP and a potential NAT box standing in the way of a direct profiling. Also we used a set of potentially unwanted programs (PUP) and their traffic logs obtained from a major security company to correlate our clients' traffic with these PUPs' activities, leading to the discovery of the RESIP's illicit operations and their providers' hidden infrastructural components.

Findings. Using our framework, we analyzed 5 leading RESIP providers including Luminati [3], Proxies Online [5], Geosurf [1], IAPS Security [2] and ProxyRack [6], from which we found 6.18 million unique IPs in a 4-month span. As a result, we were able to conduct the *first* study on RESIP service. Our analysis reveals the abused RESIPs as attack intermediaries as well as illicit and collusive RESIP service providers. Our key findings are as follows.

• Our discovered RESIPs are distributed across 238 countries and regions, 28,035 /16 network prefixes and 52,905 ISPs. A vast majority of them (95.22%) are believed to be indeed residential and very few of them (2.20%) are reported by public blacklists or emerging threat intelligence platforms.

• We discovered the presence of likely compromised hosts as RESIPs, among which, 237,029 IoT devices and 4,141 RESIP hosts running PUP programs were identified, although RESIP service providers typically claim that their proxies are all common users willingly joining their networks. In fact, none of the 5 RESIP providers is a completely consent-based anonymity system and even the most prominent companies like Luminati were found to use suspiciously compromised residential hosts.

• We identified 67 different programs running as RESIPs. Among them, 50 are reported as malicious by anti-virus tools.

• Unlike the bots as reported in prior studies [65], even the RESIPs running PUPs, as discovered in our research, exhibit very different behavior in terms of their traffic patterns, indicating new challenges in detecting them.

• We found the traffic relayed by RESIPs involves ad clicking, promotion, or malicious activities. 9.36% traffic destinations were detected as malicious by popular detection engines. Also surprisingly, we observed other monetizing services also running on the hosts of RESIPs. Examples include Fast fluxing and malicious content services.

• We observed some RESIP providers likely reselling services to (or at least sharing RESIP pools with) other providers. For example, our infiltration traffic from the IAPS proxies was actually relayed by Hola clients controlled by Luminati. We found that unlike Luminati, IAPS conducts no background check and accepts bitcoin payments. Malicious IAPS users might thus be able to abuse Luminati's network or even to cause denial-of-service for legitimate Luminati customers.

• We identified hidden backend gateways in the RESIP service infrastructure, which decouple the clients and RESIPs in their infrastructure to make illicit activities of the RESIP service stealthier: some backend gateways were labeled as malicious sites and were dropped by the providers, while all of the frontend gateways were clean and enjoyed a long lifetime.

Contributions. The contributions of the paper are as follows.

• *New findings*. Our findings revealed the infrastructure, scale, malice, and stealthiness of RESIP services. They highlight the security implications of this emerging service and the urgency to regulate its market.

• *New methodologies*. We designed novel techniques for finding RESIPs, profiling their behaviors, and analyzing the providers. They can be integrated into a holistic system for monitoring RESIPs and detecting/preventing its malicious activities.

II. Background

Residential proxy. Residential IP proxy services are a thriving business today. During our study in 2017, we continued to witness the emergence of new RESIP services and a boom in existing businesses: e.g., Proxies Online [5], the first RESIP service we found, has increased their price from $3/GB to $25/GB in 6 months. Like traditional proxy services such as virtual private network (VPN), anonymity networks, and HTTP/SOCKS proxies, RESIP service is promoted as an anonymity channel, but also characterized by its resilience against server-side detection and blocking. More specifically, residential IPs are often more trusted by the server than those from a data center [4]. Also, they tend to be dynamic, with RESIP services usually running in a back-connect proxy mode, making malicious clients nimble and capable of quickly migrating to other IPs when detected.

Figure 1 illustrates the RESIP service model discussed in the prior works [58], [59], which involves three parties interacting with each other: the main service component including a proxy

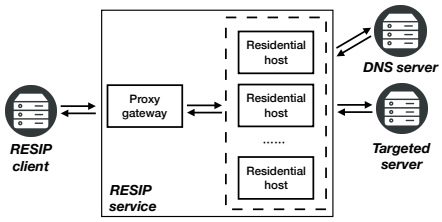


Fig. 1: The RESIP service from an outsider's perspective.

gateway and residential hosts, the client, and the server to be visited (the *target*). Once a client signs up with a RESIP service, it receives a gateway's IP address or URL for accessing the service. During the communication, the gateway forwards the client's requests to different residential hosts, which further send them to the target and get responses back. Figure 1 describes what can be observed from the outside, from the client and target's perspective. The inside view, however, can be more complicated, as discovered later in §V-B.

There are many RESIP providers on the market, such as Luminati and Geosurf. They offer a variety of service plans with different levels of flexibilities, which can be leveraged to launch cyber attacks. For example, the client is given three different ways to determine how proxies are chosen, based upon whether the gateway attempts to use the same RESIP to send multiple requests to the target: sticky (S), non-sticky (NS), and half-sticky (HS). A sticky gateway always tries to use the same RESIP for communication whenever it can, and when it has to give up on the proxy (when the RESIP gets off-line), the gateway attempts to switch to the next one. The client can also specify the "sticky time", e.g., changing to a different RESIP after 1 minute. In the non-sticky model, the gateway changes RESIP each time after a request is forwarded. The half-sticky service allows the client to switch between the S and the NS models by adjusting parameters (e.g., a session ID) during the communication. Another service option is to decide where the domain name of the target to be resolved, by the RESIP or the gateway. This is important since the resolver can be observed by the target's DNS server and may need to be covered under some circumstances. As an example, the RESIP provider Luminati allows its client to move the DNS resolving to the RESIP by using the `-dns-remote` parameter.

IP Whois Database. The Internet Assigned Numbers Authority (IANA) allocates IP addresses in large chunks to one of five Regional Internet Registries (RIRs), including ARIN, APNIC, AFRINIC, LACNIC and RIPE. Each RIR operates a Whois directory service to manage the registration of IP addresses in their regions (e.g., Europe region for RIPE). A Whois directory is organized in an object-oriented way, containing four types of objects with each assigned a unique ID: inetnum, person, organization, and ASN. Here an inetnum object describes an IP address range and all its attributes; organization and person objects are used to represent the ownership of IP blocks with a set of attributes like email addresses; and ASN identifies the autonomous system an IP address belongs to. All inetnums are created in a hierarchical manner and therefore form an inetnum tree. Given an IP, we define its *direct inetnum* as the leaf inetnum object whose IP range covers that IP, its *direct owner* as the organization and person objects associated with its direct inetnum, and its *loose owner* as all organizations and persons who share the same contact information as the direct owner. In our research, we collected the IP Whois databases from all 5 RIRs everyday since December 2015 using their RDAP and bulk access APIs [40] [46][23][24][45][44]. Those historical IP Whois databases were used to generate features for our residential IP classifier (§III-B).

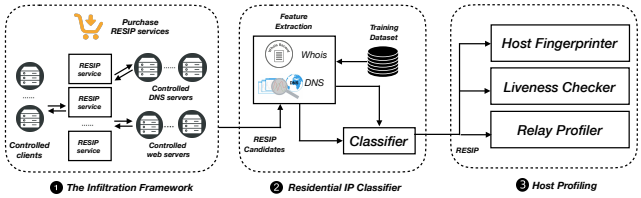


Fig. 2: Our methodology framework.

III. Methodology and Dataset

As shown in Figure 2, the methodology behind our study on RESIP consists of three important parts: an infiltration framework (§III-A) for gaining insider's views of RESIP services, a classifier (§III-B) for identifying residential IPs, and a host profiling system (§III-C) for fingerprinting the proxy hosts. We elaborate them as follows.

A. Infiltration Framework

Our infiltration framework includes a client, which is a web crawler sending *labeled* requests through a RESIP service to its target site, a target server, which is a website receiving the client's requests forwarded by RESIPs, and our own authoritative DNS server, which is utilized to find out whether DNS resolving happens on the RESIP hosts or on the gateway, and further discover these resolvers. This framework is also illustrated in Figure 2.

We found 17 RESIP services either through search engines or from Blackhat SEO forums [31]. Among them, 5 (Table I) were picked out based upon their claimed scale ($>$ 100K IPs), service models (SOCKS or not, pay by month or traffic, etc.), popularity (heavily promoted online), and the time they were discovered (earliest ones). All 5 services support relaying HTTP/HTTPS traffic and ProxyRack also supports SOCKS4 and SOCKS5 protocols. We then purchased those five RESIP services, and ran our crawler to periodically visit our server with pre-registered domains through these services. Our server recorded each labeled request and extracted its source IP, which was considered to be the address of the RESIP provided by the service. For this purpose, each request produced by our crawler was *labeled* to avoid recording the requests from other parties, since they may not carry RESIP IPs (e.g., Man in the Middle players record our traffic and replay it). Also, this approach forces the RESIP to query our DNS server, exposing its resolver. In our framework, a client sends requests to specially crafted subdomains (as part of the HTTP request URL) with the following pattern: uuid.timestamp.providerId.gwId.raap-xx.site, where uuid is a dynamically generated UUID, timestamp is the client's current Unix timestamp, providerId uniquely identifies

the RESIP service provider, gwId represents the type of the proxy gateway (S, NS or HS) and raap-xx.site represents a set of domains registered for our website, with xx describing various geo-locations (us, eu, etc.). In this way, each request targets at a unique subdomain. Moreover, such crafted requests, once being proxied by the RESIP device, became more likely to be captured by our industry partner's anomalous traffic gathering module (data collected by the module elaborated in §III-D) due to their newly registered domains carrying the patterns produced by DGA (Domain Generation Algorithms). Through such collected data, we were able to locate the RESIP devices and analyze the traffic they proxied (See §IV-C).

Upon receiving a DNS query for such a domain, our DNS server employed a regular expression to check the pattern of the subdomain, and if correct, resolved it to the IP addresses of our controlled servers. In this way, for each successful request, three log records were generated by the entities under our control: the client (our crawler), the target server, and the DNS server as illustrated in Figure 2. Here the client recorded the labeled request URL, the target server kept the RESIP's IP, and also the DNS server logged the RESIP's DNS resolver. Correlating those logs provides us a comprehensive view of a RESIP's operations, and can also help discover related traffic traces from other sources when they were captured by network monitors (see §IV). As shown in Table I, all RESIP services except Luminati resolve domain names on RESIPs rather than gateways while Luminati can do this on either site through configuration. We came to this conclusion since our DNS server received queries issued by over 82K DNS resolvers from these RESIP services in our study.

During our study, we carefully designed our methodology to ensure that our infiltration and profiling are less detectable by the RESIP services. For this purpose, we deployed multiple crawlers and target servers on Amazon EC2 instances and Aliyun instances located in European, US, South America, Singapore and China, to generate traffic from diverse sources. Further, we used AES-CBC with a 128-bit key to encrypt all traffic between our crawlers and the targets, to prevent potential content inspection. Another implementation issue is the presence of multiple gateways and the different models they are running (S, HS and NS; see §II and Table I). For example, GeoSurf and ProxyRack all run sticky gateways; as a result, our server would not see any new proxy host during a given period of time (1 to 10 minutes); therefore our crawler was implemented to only request once for a while, depending on the sticky time given by the service. For the providers with non-sticky and half-sticky gateways, our implementation took different strategies to generate requests. When there were multiple gateways, we chose a different one for each request in order to reduce redundant requests and cover more RESIPs. Besides, in case RESIP services assigned different gateways to different users, we registered for each service at least two distinct user accounts and found that each account was always linked to the same set of gateways.

Result and evaluation. In total, we ran up to 20 daily crawling jobs, each producing about 50,000 requests, from Jun. 06 to Nov. 24 2017. Our study captured 6,183,876 different RESIP IPs by issuing 62 million requests. Before Sep. 15, we only ran 2 crawling jobs on a single service, Proxies Online. Then starting from Sep. 17, we gradually purchased at least one-month service from all 5 RESIP providers and ran up to 20 crawling jobs daily using 200+ threads to collect RESIP information from all of them. After one month, we have gathered enough RESIPs from Luminati. Meanwhile, our measurement results revealed that IAPS Security was just a reseller of Luminati's service, and Geosurf and Proxies Online actually share the same infrastructure. Given the above findings, we then stopped crawling the expensive providers, including IAPS Security, Geosurf, and Luminati, but still kept the jobs on Proxies Online and ProxyRack until Nov. 24. Overall, we spent $2800 in purchasing and infiltrating those services.

Provider	Price	Payment	Date(s)	Gateway	DNS
Proxies Online	$25/Gb	Paypal	07/06-11/24	HS	R
Geosurf	$300/month	Paypal	09/17-10/22	S/HS	R
ProxyRack	$40/month	Bitcoin	09/18-11/24	S/NS	R
Luminati	$500/month	Paypal	09/25-11/01	HS	R/G
IAPS Security	$500/month	Bitcoin	09/23-11/01	HS	R

TABLE I: RESIP services purchasing details. HS: half-sticky; S: sticky; NS: non-stick; R: RESIP; G: gateway.

Source	Label	# IPs	# /16	# /8	Training
Manual	resi-clean	79	25	19	79
Device Search Engine	resi-clean	89,345	13,525	195	9,921
Trace My IP	resi-noisy	37,480	11,402	213	0
Filtered IP Whois	resi-noisy	23,264,961	394	31	0
IoT Botnets	resi-noisy	1,699,291	20,112	200	0
Public Clouds	non-resi-clean	53,716,321	968	99	5,000
Alexa Top1M	non-resi-clean	442,989	14,365	213	4,481
Commercial Proxies	non-resi-clean	519	71	44	519
Public Proxies	non-resi-noisy	148,509	14,004	204	0

TABLE II: Datasets for training and testing the residential IP classifier.

B. *Residential IP Classifier*

While RESIP service providers claim to utilize residential hosts for relaying their customers' traffic, little is known about whether the proxies they use are indeed located in residential networks. Determining whether an IP is residential can be complicated, particularly when the same ISP can also allocate IP blocks to data centers. Although some commercial service (e.g., Maxmind GeoIP2 Precision Insights Service [33]) allows queries on IP's labels such as residential or cellular for a fee (e.g., $50 for 25K IPs), it cannot scale to a large number of queries (6.2M in our research) and its methodologies are not open (so less known about their reliability). So in our research, we built a new classifier on top of a set of features that characterize residential IPs. Following we elaborate the technique, particularly, our approaches to collect clean ground truth, select robust features, and train and evaluate the classifier.

Finding groundtruth. Finding *clean* labeled residential IPs is challenging due to the absence of public data and the dynamic IP allocation performed by ISPs. To address this issue, we came up with a series of robust methodologies to obtain 4 labeled datasets: residential-clean (resi-clean), non-residential-clean (non-resi-clean), residential-noisy (resi-noisy), and non-residential-noisy (non-resi-noisy). Such groundtruth is summarized in Table II.

The *resi-clean* set contained 79 IPs of the personal devices under our control, which were connected to 11 ISPs in 3 countries for identifying these addresses. To find other "clean" IPs, we came up with an idea that leverages device search engines (e.g., Shodan [48], Zoomeye [52] etc.) to search for the network devices typically *only utilized in residential environments*. Examples include smart home systems such as Amazon Echo [27], Google Home [35], Philips Hue Lights [41], home-related gateways like residential ADSL gateway and broadband residential gateway, and others. A complete list of keywords used in such device queries is presented in Appendix IX-A. These queries return IPs for both devices discovered online and related applications. The former was added to our *resi-clean* dataset as groundtruth. In this way, we successfully harvested 89,345 residential IPs distributed across 13,525 /16 and 195 /8 network blocks. This data collection was done automatically, which we believe itself is a technical contribution.

We further applied several weaker heuristics to build the *resi-noisy* dataset. Despite being noisy, the dataset is still useful in validating our classifier. Specifically, its data comes from three sources. (1) We used the query logs of *Trace My IP* [51], an IP tracing service helping visitors to find their devices' IPs. The IPs recorded by the logs were selected as potential residential IPs when the ISPs involved are known to be residential Internet service providers (e.g., AT&T and Comcast), queries are from the OSes for consumer devices (e.g., Android and IOS) and common browsers, and the IPs are not labeled as bot or spider. (2) We looked up the owner objects for the 79 clean residential IPs in the IP Whois dataset (see § II), and considered other IPs under those owner objects as residential IPs. This is because as a common practice, ISPs (such as AT&T) typically register the same set of owner objects to manage the IP blocks serving the same purposes. For example, AT&T registers the owner object ATTMO-3 [28] for AT&T Mobility LLC [29] to manage all IPs for mobile usage. (3) We also included the IPs detected from two emerging botnet campaigns Hajime [12] and IoT Reaper [13] that utilize compromised IoT devices (see §III-D), as home IoT devices are much more likely to be compromised than enterprise IoT devices. In total, the *resi-noisy* dataset contained 25,001,529 IPs.

The *non-resi-clean* data were collected from cloud providers, high-profile websites (Alexa top 1M websites), and commercial proxies (details in Appendix IX-A). We gathered 54,031,298 such IPs distributed across 14,610 /16 and 213 /8 network blocks. The *non-resi-noisy* dataset involved the IPs from publicly available proxies (e.g., Tor relays and public free proxies) as detailed in § III-D. The data is noisy since some such proxy services like Tor also recruit home servers to relay traffic [50]. This dataset included 148,509 IPs in 14,004 /16 and 204 /8 networks.

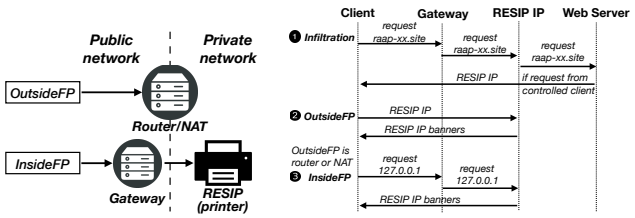
From the above datasets, we built a labeled set with 10K residential IPs and 10K non-residential IPs randomly sampled from *resi-clean* and *non-resi-clean*, respectively (see Table II). They were used in feature evaluation and classifier training while the rest datasets were applied to evaluate our classifier.

Feature selection and extraction. We selected a set of unique features to train a classifier to identify residential IPs. Unlike non-residential IPs, residential IPs are typically directly assigned and managed by an ISP (instead of being *re-assigned* to a business) [66]. Also, ISPs tend to reserve stable IP blocks (belonging to the same *inetnum*) for home users, while the network blocks given to the business could be more volatile, changing hands over multiple owners during a given period of time [66]. Furthermore, non-residential IPs are more likely to host web services. For example, among 442,989 IPs for the Alexa Top 1M domains, 29% (128,531) are found in our Public Cloud dataset while only 0.01% (36) are also in our *resi-clean* dataset. Based upon such observations, we leveraged a total 35 features related to IP Whois records or Active DNS records to capture residential IPs' characteristics. Due to the space limit, we here just elaborate some of them and the rest is presented in Appendix IX-A.

• *An Active DNS feature*. As an example, the connection between non-residential IPs and web services can be captured by the average number of TLD+3 domains per IP in the direct inetnum (§II). Intuitively, this feature describes the number of domains hosted in the direct inetnum of this IP, which were found from Active DNS dataset [68]. Our evaluation on the labeled set shows that non-residential IPs have 5.49 as the average feature value while residential IPs only have 0.016.

• *IP Whois features*. We also used phone numbers and email addresses to identify the owners of the inetnum for an IP, and discovered that residential IPs tend to have much more inetnum objects (3,536 on average) than non-residential IPs (1,482 on average). This could happen when the ISP assigns large chunks of continuous IPs to their organizational users. Additionally, we designed the features to profile the size and stability of the direct inetnum of a given IP. Specifically, we retrieved the IP's historical direct inetnums from 24 IP whois snapshots in the last 2 years, and identified their sizes, depths on the inetnum tree, and further calculated the variations of these parameters to capture their changes in the past 24 months. We observed that 70% of the residential IPs have a size (of historical direct inetnums) below 10^5, while 58% of non-residential IPs have a size above 10^5. Also residential IPs are much more stable in their depths on the inetnum tree, with a variation below 0.16.

Evaluation and results. Over 10K residential IPs and 10K non-residential IPs, we trained a Random Forest (RF) classifier, which achieved an excellent performance in a 5-Fold cross validation (precision of 95.61% and recall of 97.12%). We further evaluated the model over the four labeled datasets as well as the unlabeled dataset (6.2M RESIP IPs we collected) with sampled manual validation. Our study shows that this model made the predictions in line with the natures of these sets (more leaning toward residential or non-residential IPs in the cases of the noisy datasets) and particularly on the unlabeled set, it achieved a precision of 95.80%. When applying the model on 6.2M RESIP IPs we collected, it detected 5.9M (95.22%) residential IPs and 0.3M (4.78%) non-residential IPs. More details about the evaluation process and results can be



(a) InsideFP vs OutsideFP (b) Host fingerprinter's analysis pipeline.

Fig. 3: Host fingerprinting.

found in Appendix IX-A.

C. Host Profiling

To further understand RESIPs, it is very important to profile their host devices in addition to their IPs. As mentioned earlier, residential IPs tend to be assigned in a dynamic manner. Then, once a RESIP IP is captured, host profiling must be conducted and finished before the RESIP host has moved to another IP, otherwise, the result will be invalid. To achieve this, we designed a *real-time* profiling system that can simultaneously fingerprint newly captured RESIP hosts, measure their *relaying time* (periods when serving as RESIPs), and detect when they get offline (stop serving as RESIPs) or their IPs change. As illustrated in Figure 2, the system consists of three modules: a host fingerprinter, an IP liveness checker and a relaying time profiler, which work on a given RESIP simultaneously.

In a nutshell, the host fingerprinter will compose and send various probes to a given RESIP IP on commonly opened TCP/UDP ports including 80 for HTTP, 22 for SSH, 23 for Telnet, 443 for HTTPS, 554 for RTSP and 5000 for UPNP. Once response received and banners grabbed, the *Nmap service detection probe list* [16] will be applied to identify device type and vendor information.

This process turns out to be more complicated than it appears to be. A challenge comes from the fact that an IP can be frequently re-assigned to different hosts, often not the RESIP we are interested in. To address this problem, our profiling system immediately started fingerprinting an IP address after it was observed by our web server. This was further confirmed, in the presence of both sticky and half-sticky gateways, through sending another request right after the banners were grabbed: if the same IP was seen by our server again, we were confident that the banner belonged to the same RESIP. We call this process "outside fingerprinting" (*outsideFP*) as the probing targets at the RESIP IP from the outside. Another issue is caused by the presence of a *private network* the RESIP host often stays in. So a probe to its public IP only gets to the gateway NATs and may not reach the actual RESIP host. Our solution is based upon the observation that many RESIP providers do not inspect the target IP that the client visits, which allows our client to probe the proxy's loopback address 127.0.0.1 through its connection with the gateway. Our study found that 3 out of the 5 RESIP service providers (Proxies Online, Geosurf and ProxyRack) let this "inside fingerprinting" (*insideFP*) go through. Note that both inside and outside fingerprinting require the RESIP service running with the sticky or half-sticky gateway. Figure 3(a) illustrates these fingerprinting processes, with IoT devices (printer) being RESIPs in the private network.

To achieve a high performance when profiling a large number of IPs, our system will not conduct insideFP for a RESIP unless its outsideFP result reveals a router/NAT. This is because that insideFP has a larger request latency than the outsideFP, and is constrained by the rate limitation from RESIP service providers. If the insideFP and outsideFP cannot reach a consensus, we regard insideFP's result as the final: e.g., a RESIP was considered to be a printer when its insideFP revealed the printer and outsideFP showed a NAT. We outline host fingerprinter's analysis pipeline in Figure 3(b).

The IP liveness checker and the relay profiler scanned a given IP every 30 seconds. The former simply "pinged" the IP through typical TCP and UDP ports to find out periods when the IP was online. And the latter sent "heartbeat" requests via a connected RESIP gateway to our web servers to measure the *relaying time* of a given RESIP IP. This information also helped us improve the accuracy of RESIP fingerprinting: we consider the fingerprinting result as valid only when the relaying time of a given RESIP covers the fingerprinting period.

Evaluation and results. Running on an Amazon EC2 instance with a bandwidth of 60 Mbps, 1GB memory and one-core CPU at 2.40GHz, our system was capable of profiling 800K IPs/h, with each IP being fingerprinted in 63.57 seconds. In total, our profiling system acquired banners from 728,528 (11.78% out of 6.2 million) IPs and identified the device types and vendor information for 547,497 of them. Interestingly, 237,029 (43%) of these IPs turned out to belong to IoTs like web camera, DVR, and printer. Details of the study are in §IV-B.

D. Datasets

Our study leverages various data sources to characterize multiple dimensions of the RESIP ecosystem. Recall that by now, we have produced or used several datasets: our infiltration generated a large RESIP IP dataset (§III-A). To construct and evaluate our residential IP classifier, we collected several other datasets containing residential and non-residential IPs (§III-B); we also leveraged datasets of IP Whois and Active DNS for the classifier's feature generation (§III-B). In our host profiling framework, the Nmap service detection probe list is applied to infer devices' types (§III-C). We next elaborate other datasets to be used in our study. These datasets are jointly leveraged to characterize both individual RESIPs and RESIP services.

PUP traffic. We collaborated with our industry partner (one of leading IT companies) to utilize the PUP traffic they gathered from their customers' devices (under proper consent) from June 2017 to November 2017 for our RESIP analysis. The consent was given from the users who agreed to the terms of service when they installed our industry partner's security software. The users can revoke this consent in the software settings. Each record in the dataset logged a suspicious traffic flow (inbound and outbound) associated with a PUP they detected. For each suspicious flow, PUP's MD5, device ID, timestamp, and the flow's 5-tuple (src IP, src port, dest IP, dest port, transport-layer

protocol) are recorded, with additional information added to the 5-tuple for plaintext traffic like HTTP, and FTP. For example, for HTTP traffic, the host and (truncated) URL fields were recorded. This dataset served three purposes in our research: identifying the usage of PUPs as RESIPs, investigating RESIP traffic, and revealing the hidden infrastructural components inside the RESIP services.

Passive DNS. Another dataset we utilized is *Passive DNS* from 360 Netlab [17], which enabled us to identify Fast flux activities on RESIP IPs, and reveal the hidden infrastructural components inside the RESIP services. Each of the records includes queried domain names, time periods, their aggregated lookup volumes in the given time period.

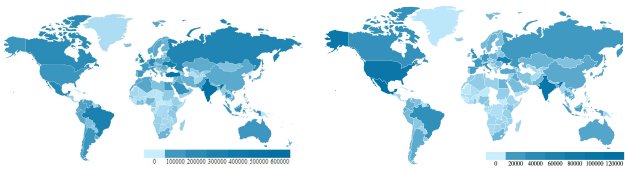
IP geolocation. IP2Location DB8 [14] is a commercial IP geolocation database provided by IP2Location. Using this dataset, we retrieved the geolocation information (country, city, latitude, longitude, ISP) for given IPs.

Public available proxies. We also collected the IPs related to public network proxies, whose traffic can be easily blocked or degraded by the server-side protection [62]. Specifically, we treated Tor relays (both exit and middle relays) as network proxies and crawled their lists hourly from both the Tor official website [19] and a third-party provider dan.me.uk [20]. We used two different ways to collect publicly available proxies for HTTP/HTTPS/SOCKS4/SOCKS5. We purchased a service called KuaiDaili, which collects proxies from multiple popular proxy aggregators [7], and provides APIs for those still working to its users. In the meantime, we also crawled other popular proxy aggregators [11] [22] to get the working proxies KuaiDaili does not include. This dataset was further complemented using IP2Proxy LITE [15], a service that runs proprietary algorithms to detect the IPs serving VPN anonymizers, open proxies, web proxies and Tor exits.

Dark IPs. Also utilized in our research are popular IP blacklists for identifying RESIP-related malicious activities. Specifically, to track the potential relation between RESIPs and two emerging botnet campaigns Hajime [12] and IoT Reaper [13], our industry partner ran a detector from Sep 15, 2017 to Nov 07, 2017 to gather bot IPs of these campaigns on a daily basis. Further, we collected 62 Spamhaus EDROP [18] records every day for the last two years. Also, APIs of three threat intelligence platforms were leveraged to retrieve IP indicators of compromise.: VirusTotal [21], Cymon OTX [10] and AlienValut OTX [9]. Given the dynamic nature of RESIPs, we only focused on IP indicators whose timestamps are consistent with those of RESIP IPs we observed.

E. Discussion

Potential bias. Due to the challenges in comprehensively identifying RESIP hosts and analyzing their illicit behaviors, our study was based upon the data we were able to get (RESIP IPs observed by our system, hosts we could fingerprint and the PUP data available to us, etc), which could bring in bias to the study. While we believe that as the first large-scale research on RESIP services, our study offers valuable insights into this new business, we are nevertheless cautious about the conclusions to be drawn. More specifically, the vantage points of our study were limited to five RESIP service providers. Also, from them, only about 10% (still more than 500K) of all the IPs we observed could be fingerprinted and analyzed. Further, our analysis on relayed traffic of RESIPs was based on the PUP traffic logs collected by our industry partner. Even though the PUP traffic logs were linked to 8,886 RESIP IPs (more than 5 millions traffic traces) in our research, their coverage is clearly limited. Availability of more comprehensive datasets will certainly help better understand RESIPs and their security implications. In the meantime, note that the RESIP providers we studied are representative and we did find PUPs running behind the RESIP IPs we could not fingerprint. This indicates that some of our results could be applied more broadly, which however needs to be determined by the future research.



(a) All RESIPs. (b) RESIPs responded to our probings.

Fig. 4: Global Distribution of RESIPs

Ethical issues. To conduct our study, we paid RESIP providers to access their services. During the study, we followed all their terms of service, and took great care to make sure that our study would not harm the owners of RESIP hosts by visiting just our own domains. Also the users of our industry partner agreed to share related information in exchange for free services. Lastly, regarding our host profiling operations, we limited probing rates to avoid overheads incurred on the remote hosts. Also we only report aggregated statistics to avoid identity leakage. All the studies were approved by our organization's IRB.

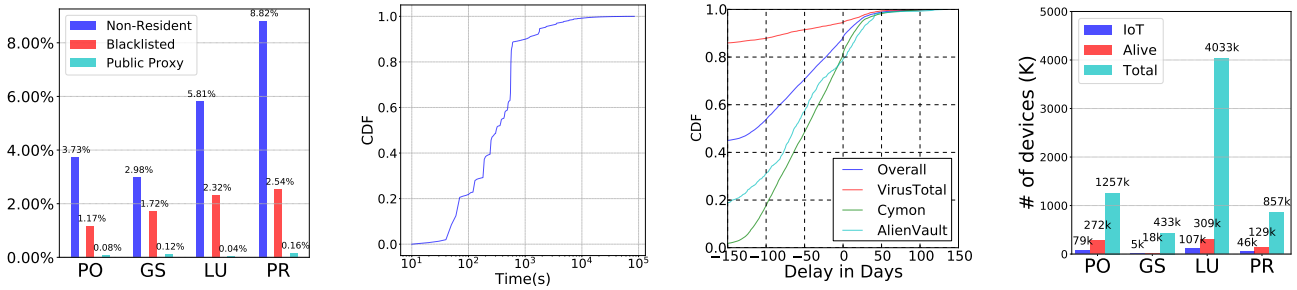
IV. Residential IP Proxy

We here report a measurement study on the core component of the RESIP service – the residential IP proxy. We analyzed why these RESIPs were used, how they were recruited, and what they served.

A. Proxy Detection Evasion

IP source analysis. In total, we collected 6,183,876 unique RESIP IPs from the five RESIP service providers via the infiltration framework (see §III-A). Our study reveals that RESIP IPs are spread across the world, across 238 countries and regions, 28,035 /16 network prefixes and 52K+ ISPs. Overall, we found that top 100 ISPs cover 57.4% of the RESIP IPs we discovered with the ISP involving most RESIP IPs being Turk Telekom (5.7%). Figure 4(a) illustrates the distribution of the RESIP IPs over countries, as determined by their geolocations. The number of RESIP IPs in each country is ranked and illustrated with various shades of darkness in the figure. As we can see here, most of RESIP IPs stay in India (9.42%), followed by Turkey (8.64%) and Ukraine (6.42%).

As described in §III-B, we trained a classifier to identify residential IPs. Figure 5(a) illustrates the percentage of non-residential IPs in each RESIP service provider. Overall, 95.22%



(a) % of non-residential, blacklisted, published proxy IPs in RESIP services

(b) The CDF of the relaying time per RESIP.

(c) Time lag of RESIPs between being blacklisted and being captured.

(d) # of IoT devices observed from each RESIP service provider.

Fig. 5: Characterizing RESIPs. In (a) and (d), PO: Proxies Online; GS: Geosurf; LU: Luminati; PR: ProxyRack.

Top 1-5	# RESIPs	%	Top 6-10	# RESIPs	%
Spam	8,299	36.55%	Malicious Sample	438	1.93%
Malicious URL	7,305	32.17%	Zombie	277	1.22%
Bruteforce	3,325	14.64%	Telnet	249	1.10%
Suspicious	629	2.77%	Trojan	171	0.75%
Dionaea	618	2.72%	EDROP	164	0.72%

TABLE III: Malicious activities related to RESIPs.

of the collected RESIP IPs are indeed residential. Also, ProxyRack was found to have the highest fraction of non-residential IPs (8.82%). Such non-residential IPs tend to be re-assigned by small ISPs to hosting providers.

We further explored the dynamics of RESIPs by examining their IPs' relaying time (see §III-C), whose cumulative distributions are presented in Figure 5(b). As we can see from the figure, a significant portion (90%) of the RESIP IPs exhibit a short relaying time (870 seconds), which renders IP-blacklist based defense on the server side less effective.

Blacklisting. We further checked whether these residential IPs were ever blacklisted, which would allow the target server to easily block them. In our study, we looked up these addresses on the IP blacklists introduced in §III-D. In total, we observed 2.20% of RESIP IPs were reported by at least one blacklist. Figure 5(a) shows the percentage of blacklisted RESIP IPs in each service provider. We found that the portion of the blacklisted RESIP IPs is fairly small. Among these services, ProxyRack has the most blacklisted RESIP IPs (2.54%), which is followed by Luminati (2.32%) and Geosurf (1.73%). When analyzing the malicious activities they were involved in, we found that spamming and malicious website hosting were two mostly reported malicious activities. Also interesting, we found that 1, 248 RESIP IPs (see Appendix IX-B) were served in two IoT botnet campaigns Hajime [12] and IoT reaper [13].

Figure 5(c) shows the cumulative distribution of the delay (in days) between when a RESIP IP was observed in our research and when it was blacklisted. We found that 11.57% of blacklisted RESIPs were captured by our infiltration framework before blacklisted, so their lifetime could be (conservatively) estimated. The average delay we observed is 22 days, with the longest being 136 days.

Unpublished proxies. When a RESIP IP is on public proxy lists such as Tor Relay list and public proxy aggregator, it can be easily blocked by the target server. To find out whether these proxies were published online, we inspected 4 proxy lists (see §III-D). The percentage of published RESIP IPs in each service provider is presented in Figure 5(a). In total, only 0.06% (3,767) of the 6.2 million RESIP IPs discovered in our research are among the 148,509 public proxies. Among all 5 providers we investigated, even the one with the most reported proxies, ProxyRack, has just 0.16% on these lists.

B. Proxy Recruitment

Volunteer recruitment. If RESIP services are recruiting volunteers, there must be related web pages and software stacks that are accessible to common users. For each service, we carefully went through their websites, read through search engine results for keywords such as *luminati recruit*, proxyrack volunteer, and *geosurf software*. Overall, only Luminati was found to explicitly recruit common users [36]. By joining Luminati's network, users can get their traffic relayed by other members at the cost of proxying others' traffic. To join the network, users need to install the hola client [30], which has versions available for multiple platforms including mobile. For other services, we found no recruitment channels or software stacks.

Fingerprinting analysis. To further explore how RESIP services recruit proxies, we analyzed devices behind RESIPs through our real-time profiling system described in §III-C.

Specifically, in our study, our profiling system acquired banners from 728,528 (11.78% out of 6.2 million) IPs observed, indicating that these were the hosts with some ports open for probing. Among these *responding* hosts, 547,497 of them returned device types identified together with their vendor information. Interestingly, 237,029 of them turned out to be IoT systems, such as web camera, DVR, and printer. Figure 5(d) presents the percentage of the IoT devices observed from each RESIP provider's network. Luminati was found to have the most IoT devices (45%), followed by Proxies Online (33%) and ProxyRack (19%).

Table IV presents the top 10 device types and top 10 vendors for the RESIPs identified. We found that most of these RESIPs (69.32%) were profiled as routers, gateways, or WAP. The manufacturers for most of the RESIP devices were MikroTik, Huawei, Technicolor, ZTE, and Dahua. Particularly, the device vendor MikroTik, Huawei, and BusyBox were associated with 59.93% of the IoT devices involved.

Note that the aforementioned result is a combination of both outside fingerprinting (outsideFP) and inside fingerprinting

(insideFP) results. As mentioned in §III-C, services including Geosurf, Proxies Online, and ProxyRack support insideFP for their sticky and half-sticky gateways. For RESIP IPs captured from those channels, insideFP was performed on a RESIP IP once its outsideFP revealed a NAT device (router, WAP, etc.). Overall, we ran insideFPs on 35,808 RESIP IPs, 12, 497 responded to our probings, and 10,964 further had their associated devices identified. Among them, 5,981, which was found to relate to gateways by outsideFP, were considered to host non-gateway devices according to insideFP. One interesting point here is that although outsideFPs on those 35,808 RESIP IPs all received responses, only 12, 497 replied to our insideFPs (using similar probings as outsideFP), indicating those unresponsive RESIP hosts may actually reside behind NAT devices. We therefore expect that the actual proportion of non-gateway devices to be higher than that in Table IV.

Also conflicting devices could be found on the same RESIP IP, particularly during host re-profiling. Re-profiling happened rarely in our study, since we did not re-profile the same IP found in 15 days. Still we observed 195 RESIP IPs hosting different devices, indicating that multiple RESIPs possibly share the same IP. Besides, even in a single fingerprinting, the banners grabbed from different ports associated with the same IP may reveal different devices. However the scenario is very rare: only 1,083 RESIP IPs (0.20% out of 547, 497) found in our study. When this happened, we simply assigned the IP most popular device identified when studying the distribution of the devices across IPs (Table IV).

One potential concern is the representativeness of our profiling results as only 11.75% RESIP IPs responded to our probings and overall 8.85% RESIP IPs had their device information identified. However, as shown in previous studies [77] [63] [64] [61], such low identification rate is quite common. For example, according to the latest large-scale probing conducted by CENSYS [43], among their probes on 0.37 billion alive IPs, only 50 million (13.5%) produced HTTP responses, 3 million (0.8%) produced TELNET responses, 10 million (2.7%) triggered FTP responses, and 13 million (3.5%) led to SSH responses, etc. Besides, as shown in Figure 4(b), RESIP IPs with devices identified are distributed globally in 215 countries and regions (16,516 /16 and 196 /8 networks). This also indicates that our host profiling results are representative.

In summary, our host profiling results indicate that rather than joining RESIP services willingly, at least some RESIP devices are likely "recruited" through stealthy compromise. On one hand, none of the five RESIP services except for Luminati provides software stacks for recruiting users. On the other hand, many IPs fingerprinted were found to host IoT devices. Although some devices like WAPs and routers may serve as the NAT front that covers other hosts behind the scene, others such as cameras, printers, DVRs and media devices, etc., are very *unlikely* to voluntarily join the services by their owners.

C. Proxy Traffic Analysis

Proxy traffic collection. In order to understand how the compromised RESIP devices operated, we leveraged the PUP

Device Type	Num	(%)	Device Vendor	Num	(%)
router	114,768	48.42	MikroTik	86,593	36.53
firewall	25,088	10.58	Huawei	37,545	15.84
WAP	24,470	10.32	BusyBox	18,337	7.74
gateway	22,003	9.28	Technicolor	16,866	7.12
broadband router	17,358	7.32	SonicWALL	14,122	5.96
webcam	13,024	5.49	Fortinet	9,190	3.88
security-misc	10,608	4.48	Dahua	6,258	2.64
DVR	4,249	1.79	ZyXEL	5,601	2.36
media device	2,589	1.09	AVM	5,272	2.22
storage-misc	1,988	0.84	Cyberoam	4,558	1.92

TABLE IV: List of the top 10 device vendors and device types.

Name	Providers	# IPs	# Devices
hola_svc.exe	LU, IAPS	2.7K	1.1K
csrss.exe	PR	241	126
svchostwork.exe	GS, PO	226	32
swufeb17.exe	PO	171	28
netmedia.exe	GS, PO	170	95
start.vbs	PO	76	1
cloudnet.exe	PR	55	42
hola_plugin.exe	LU	50	43
produpd.exe	PR	21	8
pprx.exe	PO	2	2

TABLE V: List of the top 10 PUPs with most infected RESIPs.

traffic data (see §III-D) to find the illicit activities the PUP-hosting RESIP devices were involved in. Specifically, we first analyzed the traffic logs of these PUPs, searching for the domains (those the PUP communicated with) matching the pattern of our labeled infiltration traffic. As mentioned in §III-A, the packets sent by our client to our target web server through a RESIP service were constructed in a unique way: uuid.timestamp.providerId.gwId.raap-xx.site. This labeling approach ensures that even when all other payload content of these packets was discarded, still we could identify the communication as long as the target domains were recorded. This was exactly the case for the PUP traffic logging, which only kept the domains, and another small amount of information, including the time when the communication was observed. In our study, we correlated the PUP communication with our infiltration traffic based upon the matched one-time domain, their timestamps (within 1 minute), and the log on the client side, which is supposed to record the request sent out, and the log on the server side, which should receive the request *only once*. These checks ensure that there would not be any false hit caused by, for example, traffic replay. In the end, we discovered from the PUP dataset 5,895 traffic records that accurately matched the records on our sides. Those records cover 67 different PUPs. To better understand the 67 PUPs, we scanned their MD5 using VirusTotal and found that 50 of them were flagged by at least one anti-virus engine, and each PUP on average received 24.71 alarms. We then submitted these VirusTotal reports to AVClass [75] to get the PUPs' families. In the end, 17 were labeled as cryptos, 10 as glupteba, and 5 as one of elex, bandit, zusy, wcryg and razy, and the families of the remaining PUPs were not identified.

For all these 67 PUPs, we collected their traffic logs from June 2017 to Nov 2017: totally, 5 million of them covering 8,886 RESIP IPs and 4,141 devices. Table V presents 10 PUP examples from different RESIP providers. Their MD5s are included in Table XIII of Appendix IX. The 5 million PUP

traffic logs were further used in our traffic analysis (elaborated below). Note that the above numbers are only the "lower bounds" for the pervasiveness of PUPs across RESIP services, given the limited device accesses our industry partner has.

Surprisingly, we found that all 5 services studied in our research utilized PUPs to relay traffic: 33 for ProxyRack, 9 for Luminati, 24 for Proxies Online, 10 for Geosurf and 2 for IAPS Security. Particularly, our traffic from Proxies Online and Geosurf went though 9 shared PUPs, which together with other findings (see §V-B) indicates that these services are likely all affiliated with the same company. Also surprisingly, the proxy program used by Luminati, *Hola*, was marked as PUPs, and some of them (2 out of 9) were forwarding our infiltration traffic sent to a different RESIP provider, IAPS. This combined with further analysis in §V-B indicates that IAPS is very likely a reseller for Luminati's RESIP service.

Traffic Target analysis. Our access to the PUP traffic log helped us learn more about other illicit activities performed by RESIPs. Specifically, from the 5-million traffic logs of 67 PUPs, we extracted destination domains, URLs and IPs of their communication, as well as related traffic volume. Manual analysis of top 1,000 destinations with the largest traffic volume shows most of them reside in the following 5 categories: ad (75%), searching engines (8%), shopping (7%), malicious websites (5%) and social networks (2%). Among ads-related domains, the majority are affiliate networks such as tracking.sumatoad.com, click.howdoesin.net, www.alexacn.cc, and click.gowadogo.com. Others are dedicated to different ad services such as mobile advertising, in-app advertising, video advertising, ad exchanges. Many of those ad domains are reported to install adware on users' devices such as ads.stickyadstv.com, counter.yadro.ru, and adskpak.com. Those adware altered browser homepages, generated various forms of ads. Further, analysis of corresponding URLs of those domains shows that most of them are in the forms of ads provided by those domains. Examples include click.howdoesin.net, tracking.sumatoad.com/aff_c?, click.gowadogo.com/click? and proleadsmedia.afftrack.com/click?. We also observed lots of search queries are sent to different search engines including Google Search, Bing Search, Baidu Search, Yandex, and also visits to various shopping websites including amazon.com, ebay.com, sears.com and tmall.com. Given that those proxy services are rather expensive, with 1 GB costing at least $15, using them for daily shopping and online search does not seem to be reasonable. More likely were the activities related to blackhat SEO or other online promotion operations. What is more, some websites such as lenzmx.com and csgob0t.online were found to be malicious in our manual analysis, in line with the results reported by VirusTotal.

Further we found from the PUP logs the traffic to known malicious domains. Specifically, 9.36% of the destination addresses were reported to be malicious by VirusTotal (68.92% are labeled as malware sites, 29.97% being malicious sites and 2.24% being phishing sites). Examples include ntkrnlpa.cn, gwf-bd.com, fadergolf.com, www.2345jiasu.com, and www.pf11.com, which have been reported by the most detection engines on VirusTotal like Google Safebrowsing, BitDefender, CLEAN MX, etc.

Domain	Usage	# RESIPs	# Subdomains
noip.com/ddns.net	Dynamic DNS provider	217	225
opengw.net	P2P VPN	206	509
Hopto.org	Dynamic DNS provider	54	73
no-ip.biz	Dynamic DNS provider	35	172
duckdns.org	Dynamic DNS provider	28	42

TABLE VI: List of the top 5 domains resolved to most RESIP IPs.

Fast fluxing. Also surprisingly, we discovered that RESIPs serve as Fast flux proxies for malicious websites to evade IP based detection. In a fast flux, numerous IP addresses associated with a malicious domain are swapped in and out with high frequency. Applying Passive DNS data and VirusTotal APIs to the sampled 600K RESIPs, we discovered that 1.14% of the proxy IPs were once mapped to malicious domains during the periods when they were RESIPs, and on average, the mapping from these malicious domains to the proxy IPs lasted 86.8 days. However, the median was only 2 days. Table VI lists the top 5 domains resolved to most proxy IPs. Except for opengw.net which allows volunteers to serve as VPNs for others, all other four are dynamic DNS providers. Some of them are previously reported being abused by the miscreant to conduct various illicit activities [8], which are also confirmed by us, as many subdomains of them are labeled by VirusTotal as malicious such as yohoy.no-ip.biz, darkjabir.no-ip.info, and 595685744.duckdns.org.

D. RESIP vs. Bots

Another interesting question is how RESIPs relate to bots, especially, whether RESIPs are bots, and whether methodologies for detecting bots work for RESIPs. Regarding whether RESIPs are bots, we identified connections between them. In particular, 1,248 IPs were blacklisted as bots of Hajime or IoT Reaper on the same day when they offered proxy services (see Appendix IX-B); in addition, we also identified devices that were likely recruited through stealthy compromise, as detailed in §IV-B. Both indicate the existence of bots acting as RESIPs. Nevertheless, we also identified channels for volunteer recruitment, suggesting willingly joined users are also part of the RESIP networks.

Meanwhile, compared to bots, RESIPs are observed to exhibit different characteristics that indicate new challenges for detection. Unlike a bot, a RESIP is a proxy to help users access web services in a seemingly legitimate way. Although RESIP services recruit hosts in a highly suspicious manner, they likely also include legitimate volunteer participants. A prominent example is Luminati, which has a recruitment system. Furthermore, identified RESIP programs, including the PUPs, all have limited privileges, while bots usually acquire the highest privilege [74]. Also, unlike the botnet exclusively serving cybercrimes, RESIP services are promoted publicly and are likely also utilized by legitimate users. In addition, botnets are found to flux the addresses (IPs and domains) of their C&C servers or run them on bulletproof hosting to evade detection and blocking [76][54]. In contrast, RESIP services only involve a limited number of server IPs and domains, and most of them belong to popular hosting providers (See §V-B).

Source (# Machine Hours)	Flows	IPs	Ports	IP-Ports
Bots (241)	1,365.97	328.34	10.12	330.40
Normal (461)	762.38	30.41	6.41	37.44
RESIPs (64,833)	96.37	53.54	6.27	58.59

TABLE VII: Comparison of bots, normal hosts and RESIPs. All the statistics here are averaged over the number of machine hours.

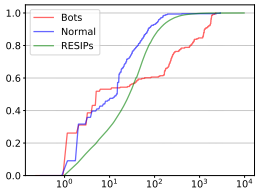


Fig. 6: CDF of # of (IP, Port) pairs visited each machine hour

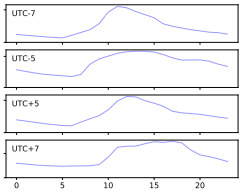


Fig. 7: # of RESIPs in each local hour of various time zones.

Therefore, intuitively the collective behaviors of a RESIP service can be very different from these of a botnet, which was confirmed by our study based on the RESIP traffic logs (§III-D) and a representative botnet traffic dataset (CTU-13 [65]) with the network flows of both normal hosts and 7 different types of bots. In the study, we looked at the network flow features commonly used for botnet detection [57] [84] [82] [67] . Examples include unique flows per machine hour, unique destination IPs per machine hour, and unique destinations (IP/Port pairs) per machine hour. Figure 6 illustrates the CDFs of the unique destinations visited every machine hour by bots, normal hosts and RESIPs: compared to the bot traffic, the RESIP traffic looks more similar to the normal one, as also observed when comparing other features across the RESIP and botnet datasets (Table VII). This indicates that the mixture of legitimate and illicit traffic of the RESIP service moves its statistical features closer to these of the legitimate communication. Despite the above findings, we must acknowledge the limitations of our approaches. For example, we are not able to exhaustively consider all bot and RESIP types; the traffic data containing only the network flow information does not allow us to experiment detection methodologies such as those based on deep packet inspection (DPI). Therefore, we leave more detailed comparison analysis between RESIPs and bots as our future work.

V. The RESIP Ecosystem

A. Landscape of RESIP Service

Through infiltrating RESIP services, we were able to collect a pool of RESIP IP addresses. Specifically, everyday during the infiltration period, we launched multiple RESIP crawling jobs running across different hours in the whole day from different locations and accounts, trying to reveal the landscape of the RESIP pool. Overall, we captured 6 million RESIP IPs by sending 62 million requests. Note that due to the IP churn issue especially in mobile networks, the number of RESIP IPs here should only be considered as an upper bound of the number of RESIP hosts. Table VIII shows the RESIPs distribution in different network blocks and ASes for each RESIP service provider. We can observe that Luminati has the largest RESIP pool, followed by Proxies Online and ProxyRack.

Table IX lists the top 3 countries, ASNs and ISPs with most RESIPs. They all exhibit long-tailed distributions where

Provider	# RESIP	# /24	# /16	# /8	# ASN
Proxies Online	1,257,418	483,310	19,654	196	7,701
Geosurf	432,975	221,747	15,143	194	4,971
ProxyRack	857,178	345,648	19,520	196	8,751
Luminati	4,033,418	1,183,841	22,467	197	17,820

TABLE VIII: Distribution of RESIPs.

Provider	Top Countries	%	Top ISPs	%	Top ASNs	%
Proxies Online	India	32.2	BSNL	6.5	9829	8.1
	USA	7.8	Uninet S.A. de C.V.	5.2	8151	5.4
	Mexico	6.7	Deutsche Telekom AG	2.8	24560	4.9
Geosurf	India	27.9	Uninet S.A. de C.V.	6.9	8151	7.2
	Brazil	9.2	BSNL	4.7	9829	5.8
	Mexico	9.1	Deutsche Telekom AG	2.8	55836	4.5
ProxyRack	Russia	8.6	PT Telkom Indonesia	5.4	17974	5.3
	Indonesia	8.1	Pakistan Telecom	3.7	8452	4.7
	Egypt	6.3	Republican Unitary	3.3	45595	4.0
Luminati	Turkey	12.7	Turk Telekom	8.5	9121	8.5
	Ukraine	7.9	JSC Ukrtelecom	1.7	25019	1.8
	UK	6.1	BT	1.7	34984	1.8

TABLE IX: Top 3 countries, ASNs and ISPs with most RESIPs

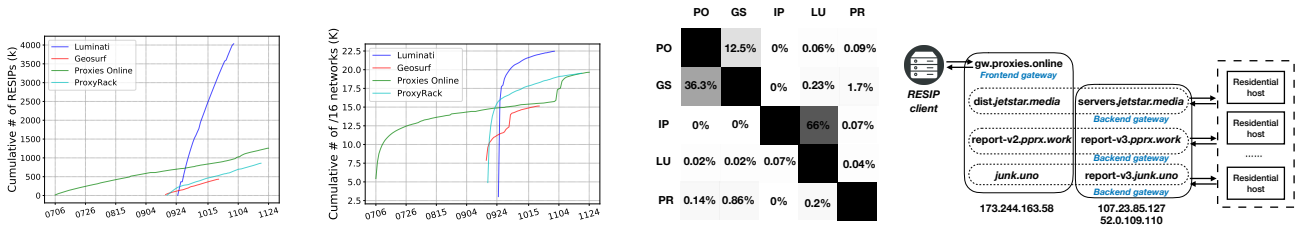
a small fraction of countries, ASNs and ISPs contribute the majority of RESIPs, respectively. For example, we find that even though Luminati is located in the United States, most of its RESIPs are from Turkey, possibly because of Turkey's network censorship which makes Hola clients a good option to visit blocked websites there. An interesting finding here is that despite Luminati's claim of having 30 million IPs, we only found 4 millions using 16-million probings. It is unclear where this gap comes from.

We also measured how many RESIPs a time zone contributes during its different local hours. As shown in Figure 7, the peak hours across time zones indeed exhibit diurnal patterns, confirming our previous findings that the majority devices of RESIPs are indeed residential hosts that are more likely to be powered off or disconnected during the night.

Figure 8(a) shows the evolution of the RESIP pools by plotting the cumulative number of unique RESIP IPs. We observe that a large number of RESIP IPs newly appear every day with an average increase rate of 44%. However, when considering the increase of fresh /16 IP prefixes, we observe a much smaller rise (11%) in Figure 8(b). This is reasonable because a given RESIP host is less likely to migrate from one /16 IP prefix to another than to change from one IP to another.

B. Infrastructure and Service

Backend (hidden) gateways. Under the known infrastructure of the RESIP service as illustrated in Figure 1, we found that there are a series of hidden backend servers intermediating between the frontend gateways and RESIPs, as shown in Figure 8(d). Since those servers can be regarded as gateways from the perspective of RESIPs, we call them backend (hidden) gateways. These gateways were discovered from the connections between the proxy gateway and the RESIP, as documented by our traffic logs, PUP traffic, and Passive DNS datasets. Specifically, using Proxies Online as an example, we observed that before relaying our infiltration traffic, the PUP-hosted RESIPs always communicate with lb-api.lambda.servers.jetstar.media, report-v3.pprx.work, or report-v3.junk.uno instead of



(a) Cumulative number of RESIPs. (b) Cumulative number of /16 RESIPs. (c) RESIP IP overlap between different service providers. (d) Build up the connection between the frontend gateways and backend gateways.

Fig. 8: The evolution of RESIP pools (a)(b) and the collusion of the service providers (c). In (c), "PO" stands for Proxies Online; "GS" stands for Geosurf; "IP" stands for IAPS; "LU" stands for Luminati; "PR" stands for ProxyRack.

Provider	Frontend gateway	Backend gateway
Proxies Online	gw.proxies.online	servers.jetstar.media; pprx.work; junk.uno
Geosurf	gw1.geosurf.io	servers.jetstar.media; pprx.work; junk.uno
Luminati	zproxy.luminati.io	zserver.hola.org

TABLE X: Frontend and backend gateways of RESIP services.

gw.proxies.online, which is the frontend gateway. We then investigated the PassiveDNS and found that the subdomains of jetstar.media, pprx.work, junk.uno, and proxies.online share a set of IPs as shown in Figure 8(d). This strongly indicates that jetstar.media, pprx.work, and junk.uno also belong to Proxies Online, and some of its subdomains act as backend gateways to communicate with the RESIPs. Table X lists the hidden backend gateways obtained from PUP traffic for all providers. Interestingly, we found that some hidden backend gateways (pprx.com) were labeled by VirusTotal as malicious sites (at least three indicators) while all of the frontend gateways were clean. This indicates that decoupling different components actually makes the ecosystem more robust.

Collusion. The study of RESIP traffic in §IV-C reveals that RESIP service providers Proxies Online and Geosurf shared 9 PUPs. Here we further explore the relations among different RESIP service providers in terms of their shared RESIPs. We calculate the intersection rate ($\frac{|A \cap B|}{|A|}$) between the RESIPs captured from different service providers, and further define a very strict criterion to decide whether a RESIP can be considered as shared by two providers. Specifically, we consider a RESIP as shared only if it has ever been captured in the same hour by independent infiltrations on both providers' services. As shown in Figure 8(c), we found a number of RESIPs spanning different RESIP service providers. The most popular one, Luminati, share 813 RESIPs with Proxies Online, 983 with Geosurf, 2,783 with IAPS Security, and 1,718 with ProxyRack. Besides, given that Proxies Online and Geosuf share a large portion of their RESIPs, they are likely two brands of the same company, while IAPS is probably a reseller of Luminati as most of its RESIP IPs come from Luminati.

Infrastructure Profiling. After identifying the infrastructure of RESIP services including the frontend websites/gateways and the backend gateways, we conducted further profiling to find the potential features for detecting those infrastructures. For this purpose, we first collected the IPs associated with those infrastructures by sending DNS queries from multiple locations to 48 identified domains and got 915 IPs. Then we ran periodic port scanning on those IPs and found that those frontend and backend gateways tend to open lots of consecutive ports. Specifically, Luminati has 23000-23999, 52225 and 52951 ports opened for frontend gateways and 6861-7009 for backend gateways. Geosurf/Proxies Online have 8010-8237 for frontend gateways and 11211 for backend gateways. Also, ProxyRack opens 1200-1250 and 1500-1750 for frontend gateways. We also randomly scanned the IPs of popular web services and found that none of them open such unusual ports. These ports are related to different proxy services provided by PrxoxyRack and Geosurf/Proxies Online. However, we do not know how Luminati uses those consecutive ports.

C. Case Study: Luminati

Luminati claimed to be a network where users join willingly by installing client software such as browser extensions or Hola VPN, in order to contribute their network resources while enjoying traffic relaying through other participants. Actually, when we purchased their service, Luminati indeed performed a background check that asked for photo ID and explained to us their traffic policy through a video chat (although only crawling Google is stated to be forbidden). Surprisingly, we found that Luminati (1) proxies through IoT devices that do not support Hola client software, (2) likely resells services to other providers such as IAPS that conduct no background check, and (3) involves RESIPs that host malicious content or are associated with suspicious domains. Specifically, leveraging our IP profiling infrastructure as described in §III-A, we performed a real-time device fingerprinting for newly captured RESIPs from Luminati, and identified lots of IoT devices associated with Luminati's RESIPs like webcam (4.31%), DVR (1.93%), printer (0.13%), VoIP (0.09%) and NAS (1.24%). As Luminati did not provide any Hola clients for these types of devices, our findings undermine its claim to be a network consisting of only willing participants. Instead, IoT devices appear to be an important RESIP source of Luminati.

Our findings in §IV-C and §V-B indicate that IAPS likely resells Luminati's RESIP service: the PUP traffic logs show that our infiltration traffic from the IAPS proxies was actually relayed by the Hola clients believed to be controlled by Luminati; further, 66% of the RESIPs captured from IAPS were also discovered by our infiltration targeting Luminati during the same hour. We found that IAPS conducts no background

check, accepts various payment methods such as bitcoin, and applies no traffic restrictions. Therefore, IAPS users might be able to abuse Luminati's network, or even to deny the services for legitimate Luminati customers. We also found that 2.32% of Luminati's RESIPs were hosting malicious content or having suspicious domains resolved to them while acting as proxies. Examples of such domains include the scam site tummytickle.com and the drive-by-download site www.iwys.cc, and malicious samples downloaded from those RESIPs include PUP, Trojan and exploit code.

VI. Discussion

Mitigation. Our measurements have identified numerous security issues including compromised devices and abusing RESIP services for malicious activities. A key prerequisite for mitigating such security issues is effective detection of RESIP services and RESIPs, which we plan to pursue as future work. We discuss potential features that are useful for detection.

We first consider detecting RESIP services. We propose to detect three components: their websites, frontend gateways, and backend gateways. (1) Based on our experiences, RESIP websites typically contain noticeable keywords such as "residential IP", "never blocked" and "HTTP/HTTPS/SOCKS", which can be used by a search engine or forum crawler for automated content analysis. (2) Frontend gateways are oftentimes co-located with RESIP websites with the same domain names or even IP addresses. Furthermore, as described in §V, frontend gateways tend to open a large number of TCP ports to serve traffic with various proxy requirements. This feature can also be leveraged as well for detection. (3) Several features can be possibly leveraged to detect backend gateways: opening a large number of TCP ports, having globally distributed sources of DNS queries for a low-reputation domain, and being co-located directly or indirectly with the frontend gateways.

Detecting RESIPs seems challenging. Their discovery can be facilitated using the detected backend gateways as "step stones", since RESIPs have to communicate with the backend gateways. Besides, the visiting patterns and targeted domains of traffic relayed by RESIPs may deviate from those of normal traffic, and can possibly be considered by a detection scheme.

Datasets and Code release. We will release related datasets and source code, as detailed in Appendix IX-C.

VII. Related Work

Dark Web Proxy. The security issue on web proxy services is attracting increasing attention from researchers. In particular, Weaver et al. [81] conducted a measurement study to understand the purpose of free proxy services based on how they modify traffic. Chung et al. [58] studied a paid proxy service to uncover content manipulation in end-to-end connection. O'Neill et al. [72] measured the prevalence of TLS proxies and identified thousands of malware intercepting TLS communications. Carnavalet et al. [62] released security vulnerabilities in TLS proxies, allowing attackers to mount man-in-the-middle attacks. Recently, [80] and [73] showed the content modification behavior of Open HTTP proxy services and free HTTP/HTTPS proxy services. In contrast to the above studies on web proxies and content manipulation, our research study an emerging online gray business RESIP service, and focus on the abused RESIPs as attack intermediaries and collusive RESIP service providers.

Compromised Host Detection. How to detect compromised host has been studied for long. Techniques have been developed to analyze web content, redirection chains, and traffic pattern. Examples of the content-based detection include a system [56] monitoring the evolution of web content to identify an infection using signatures generated from such modifications, and a framework [70] conducting semantic differential analysis to identify the infection of the website. Other studies focus on malicious redirectors and attack infrastructures. Examples include *JsRED* [69] that used a differential analysis to automatically detect malicious redirect scripts hosts, and *Shady Path* [79] that captured a compromised host by looking at its redirection graph. Also, a large number of studies detected compromised hosts using traffic analysis via active or passive probing. [71] detected P2P bots by remotely probing the hosts and analyzing the response traffic. [83] combined binary analysis and traffic analysis for P2P bot detection. In our study, we perform best-effort identification and characterization of RESIPs using novel methods. We also compare RESIPs to other types of compromised hosts such as bots, and reveal several challenges for accurately detecting the RESIPs on today's Internet.

Empirical study of botnet. Botnets have long been studied. For example, [53] revealed structural and behavioral features of botnets such as the high churn rate within a botnet. [60] studied the relationship between botnet and spamming activities. [78] characterized the personal data theft behavior of the Torpig botnet. In contrast, our study focuses on RESIP services that show different characteristics from botnets in their hosts, users and network behaviors, as detailed in §IV-D.

VIII. Conclusion

RESIP service is an emerging online gray business, whose security implications have never been studied before. In the paper, we report the first systematic research on this new service, based upon a suite of techniques that address the challenges in collecting RESIP host information and finding illicit activities these proxies are involved in. Specifically, through infiltrating 5 representative services, we gathered over 6.2 million RESIP IPs and further successfully profiled more than 500K hosts, identifying more than 200K IoT devices likely to be compromised to serve as proxies. Further by linking the IPs to the PUP traffic data provided by our industry partner, we gained a rare look inside the operations of these residential proxies. Our study shows that RESIPs tend to be part of such illicit activities as blackhat SEO, Fast fluxing, phishing, malware hosting, etc. Our infiltration analysis also discovered the hidden layer of their infrastructure and the collusions across different services. Moving forward, we believe that unregulated RESIP services indeed pose new threats to the Internet users and further research is needed to get a more comprehensive view of the services and develop effective solutions to mitigate their security risks.

ACKNOWLEDGMENT

We are grateful to our shepherd Professor Matthew Smith and the anonymous reviewers for their insightful and helpful comments. The IU authors are supported in part by NSF 1408874, 1527141, 1618493, 1618898 and ARO W911NF1610127. Also, authors from Tsinghua University are supported in part by the National Natural Science Foundation of China (grant 61772307) and CERNET Innovation Project NGII20160403.

REFERENCES

[1] Geosurf: Residential and data center proxy network. https://www.geosurf.com/.
[2] Iaps security. https://www.intl-alliance.com/.
[3] Luminati: largest business proxy service. http://luminati.io/.
[4] The netflix vpn ban can be bypassed – here's how it can be done responsibly.
[5] Proxies online. http://proxies.online.
[6] Proxyrack. https://www.proxyrack.com/.
[7] Public proxy service. www.kuaidaili.com/.
[8] On the trail of malicious dynamic dns domains. https://umbrella.cisco.com/blog/2013/04/15/on-the-trail-of-malicious-dynamic-dns-domains/, 2013.
[9] Alienvalut otx. https://otx.alienvault.com, 2017.
[10] Cymon otx. https://cymon.io/, 2017.
[11] Free proxy list. http://www.freeproxylists.com, 2017.
[12] Hajime - netlab opendata project. http://data.netlab.360.com/hajime/, 2017.
[13] Iot reaper: A rappid spreading new iot botnet. http://blog.netlab.360.com/iot_reaper-a-rappid-spreading-new-iot-botnet-en/, 2017.
[14] Ip2location db8. https://www.ip2location.com/databases/db8-ip-country-region-city-latitude-longitude-isp-domain, 2017.
[15] Ip2proxy lite. https://lite.ip2location.com/database/px1-ip-country, 2017.
[16] Nmap service detection probe list. https://svn.nmap.org/nmap/nmap-service-probes, 2017.
[17] Passive dns from 360 netlab. https://passivedns.cn, 2017.
[18] Spamhaus edrop. https://www.spamhaus.org/drop/, 2017.
[19] Tor exit nodes. https://check.torproject.org/exit-addresses, 2017.
[20] Tor node list from dan. https://www.dan.me.uk/tornodes, 2017.
[21] Virustotal. https://www.virustotal.com, 2017.
[22] Webanet free proxy list. https://webanetlabs.net/publ/24, 2017.
[23] Acess to apnic whois data. https://www.apnic.net/manage-ip/using-whois/bulk-access/, 2018.
[24] Afrinic bulk whois data. https://www.afrinic.net/library/membership-documents/207-bulk-whois-access-form-, 2018.
[25] Aliyun ip ranges. https://ipinfo.io/AS37963, 2018.
[26] Amazon aws ip address ranges. https://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html, 2018.
[27] Amazon echo. https://en.wikipedia.org/wiki/Amazon_Echo, 2018.
[28] At&t mobility llc. https://whois.arin.net/rest/org/ATTMO-3, 2018.
[29] At&t mobility llc. https://en.wikipedia.org/wiki/AT%26T_Mobility, 2018.
[30] Available hola clients. https://hola.org/download, 2018.
[31] Blackhat seo forum: Proxies for sal. https://www.blackhatworld.com/forums/proxies-for-sale.112/, 2018.
[32] Cloudflare ip ranges. https://www.cloudflare.com/ips/, 2018.
[33] Geoip2 precision insights service. https://www.maxmind.com/en/geoip2-precision-insights, 2018.
[34] Google compute engine ip ranges. https://cloud.google.com/compute/docs/faq#where_can_i_find_product_name_short_ip_ranges, 2018.
[35] Google home. https://en.wikipedia.org/wiki/Google_Home, 2018.
[36] Hola faq. https://hola.org/faq#intro-cost, 2018.
[37] Ibm cloud ip ranges. https://console.bluemix.net/docs/infrastructure/hardware-firewall-dedicated/ips.html#ibm-cloud-ip-ranges, 2018.
[38] Microleaves. https://microleaves.com/, 2018.
[39] Microsoft azure datacenter ip ranges. https://www.microsoft.com/en-us/download/details.aspx?id=41653, 2018.
[40] Obtaining bulk whois data from arin. https://www.arin.net/resources/request/bulkwhois.html, 2018.
[41] Philips hue lights. https://en.wikipedia.org/wiki/Philips_Hue, 2018.
[42] Pure vpn. https://www.purevpn.com/, 2018.
[43] Raw scan data of censys. https://censys.io/data, 2018.
[44] Rdap protocol. https://about.rdap.org/, 2018.
[45] Request for bulk whois of lacnic. http://www.lacnic.net/en/web/lacnic/manual-8, 2018.
[46] Ripe whois apis. https://www.ripe.net/analyse/archived-projects/ris-tools-web-interfaces/riswhois, 2018.
[47] Salesforce ip ranges. https://help.salesforce.com/articleView?id=000003652&type=1, 2018.
[48] Shodan. https://www.shodan.io/, 2018.
[49] Storm proxies. http://stormproxies.com/, 2018.
[50] Tor volunteer. https://www.torproject.org/getinvolved/volunteer.html.en, 2018.
[51] Trace my ip. http://www.tracemyip.org/, 2018.
[52] Zoomeye. https://www.zoomeye.org/, 2018.
[53] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 41–52. ACM, 2006.
[54] S. Alrwais, X. Liao, X. Mi, P. Wang, X. Wang, F. Qian, R. Beyah, and D. McCoy. Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 805–823. IEEE, 2017.
[55] M. Antonakakis, T. April, M. Bailey, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, D. Menscher, C. Seaman, N. Sullivan, et al. Understanding the mirai botnet. 2017.
[56] K. Borgolte, C. Kruegel, and G. Vigna. Delta: automatic identification of unknown web-based infection campaigns. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 109–120. ACM, 2013.
[57] L. Carl et al. Using machine learning technliques to identify botnet traffic. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on. IEEE*, 2006.
[58] T. Chung, D. Choffnes, and A. Mislove. Tunneling for transparency: A large-scale analysis of end-to-end violations in the internet. In *Proceedings of the 2016 ACM on Internet Measurement Conference*, pages 199–213. ACM, 2016.
[59] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. A longitudinal, end-to-end view of the dnssec ecosystem. 2017.
[60] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane. Using uncleanliness to predict future botnet addresses. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 93–104. ACM, 2007.
[61] A. Cui and S. J. Stolfo. A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 97–106. ACM, 2010.
[62] X. d. C. de Carnavalet and M. Mannan. Killed by proxy: Analyzing client-end tls interception software. In *Network and Distributed System Security Symposium*, 2016.
[63] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman. A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 542–553. ACM, 2015.
[64] Z. Durumeric, E. Wustrow, and J. A. Halderman. Zmap: Fast internet-wide scanning and its security applications. In *USENIX Security Symposium*, volume 8, pages 47–53, 2013.
[65] S. Garcia, M. Grill, J. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *computers & security*, 45:100–123, 2014.
[66] E. J. Hernandez-Valencia. Architectures for broadband residential ip services over catv networks. *IEEE Network*, 11(1):36–43, 1997.
[67] P. Kalaivani and M. Vijaya. Mining based detection of botnet traffic in network flow.
[68] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, and R. Joffe. Enabling network security through active dns datasets. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 188–208. Springer, 2016.
[69] Z. Li, S. Alrwais, X. Wang, and E. Alowaisheq. Hunting the red fox online: Understanding and detection of mass redirect-script injections. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 3–18. IEEE, 2014.
[70] X. Liao, K. Yuan, X. Wang, Z. Pei, H. Yang, J. Chen, H. Duan, K. Du, E. Alowaisheq, S. Alrwais, et al. Seeking nonsense, looking for trouble: Efficient promotional-infection detection through semantic inconsistency

search. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 707–723. IEEE, 2016.
[71] A. Nappa, Z. Xu, M. Z. Rafique, J. Caballero, and G. Gu. Cyberprobe: Towards internet-scale active detection of malicious servers. In *In Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS 2014)*, pages 1–15, 2014.
[72] M. O'Neill, S. Ruoti, K. Seamons, and D. Zappala. Tls proxies: Friend or foe? In *Proceedings of the 2016 ACM on Internet Measurement Conference*, pages 551–557. ACM, 2016.
[73] D. Perino, M. Varvello, and C. Soriente. Proxytorrent: Untangling the free http (s) proxy ecosystem. 2018.
[74] D. Plohmann, E. Gerhards-Padilla, and F. Leder. Botnets: Detection, measurement, disinfection & defence. *European Network and Information Security Agency (ENISA)*, 1(1):1–153, 2011.
[75] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero. Avclass: A tool for massive malware labeling. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 230–253. Springer, 2016.
[76] S. Soltani, S. A. H. Seno, M. Nezhadkamali, and R. Budiarto. A survey on real world botnets and detection mechanisms. *International Journal of Information and Network Security*, 3(2):116, 2014.
[77] D. Springall, Z. Durumeric, and J. A. Halderman. Ftp: The forgotten cloud. In *Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on*, pages 503–513. IEEE, 2016.
[78] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: analysis of a botnet takeover. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 635–647. ACM, 2009.
[79] G. Stringhini, C. Kruegel, and G. Vigna. Shady paths: Leveraging surfing crowds to detect malicious web pages. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 133–144. ACM, 2013.
[80] G. Tsirantonakis, P. Ilia, S. Ioannidis, E. Athanasopoulos, and M. Polychronakis. A large-scale analysis of content modification by open http proxies. 2018.
[81] N. Weaver, C. Kreibich, M. Dam, and V. Paxson. Here be web proxies. In *International Conference on Passive and Active Network Measurement*, pages 183–192. Springer, 2014.
[82] U. Wijesinghe, U. Tupakula, and V. Varadharajan. An enhanced model for network flow based botnet detection. In *Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015)*, volume 27, page 30, 2015.
[83] Z. Xu, L. Chen, G. Gu, and C. Kruegel. Peerpress: utilizing enemies' p2p strength against them. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 581–592. ACM, 2012.
[84] H. R. Zeidanloo, A. B. A. Manaf, R. B. Ahmad, M. Zamani, and S. S. Chaeikar. A proposed framework for p2p botnet detection. *International Journal of Engineering and Technology*, 2(2):161, 2010.

IX. Appendix

A. Residential Classifier

Crafted residential device names and types. The crafted residential device names and types are listed in Table XI. They are either consumer devices exclusively used in home network environment or network function devices usually working as components of residential network facilities.

Device Names	Phillips Hue Light
	Amazon Echo
	Wemo Switch
	Nest Thermostat
	Amazon Fire TV
Device Types	Broadband Residential Gateway
	Residential ADSL Gateway
	VoIP Phone Adapter
	Media Device
	DVR

TABLE XI: Crafted residential device names and types

Sources of non-residential ground truth Here we provide more details about our non-residential datasets as introduced in §III-B. To collect IPs from cloud services, we gathered lists of IP CIDRs published by popular cloud providers including Amazon AWS [26], Google Cloud [34], Microsoft Azure [39], IBM Cloud [37], Aliyun [25], CloudFlare [32], and Salesforce [47]. All those together contribute 53-million IPs distributed in 210K /24 and 968 /16 network blocks. We further looked up the Active DNS database for Alexa top 1 million websites and gathered 442K IPs. Another 519 IPs are collected from PureVPN[42], a popular commercial VPN service.

Features Before going through all 35 features, let's firstly refresh you the following definitions (introduced in §II) used in our features. For each IP address, we define *Direct Inetnum* as the leaf inetnum node where this IP resides in, *Inetnum Tree Path* as the inetnum path from the root inetnum node(0.0.0.0/0) to its Direct Inetnum. We also define two kinds of owners, one is *Direct Owner* represented by the organization ID or person ID referred in its direct inetnum, the other is *Loose Owner* represented by all org and person objects sharing with the direct owner the same contact information including either phone numbers or email addresses. As introduced in §III-B, 35 features are introduced in our residential classifier and they can be grouped into two categories by the datasets used to generate them: IP Whois and Active DNS.

Features from Active DNS. We retrieve DNS records from the latest ActiveDNS database for the following targets: the given IP, its current direct inetnum, its /24 IP prefix. Then, we profile each target using TLD+2/TLD+3 domains resolved to the IP range of the target. Specifically, we designed the following 12 features.

- *F-1*: # of TLD+2 domains resolved to the given IP.
- *F-2*: # of TLD+3 domains resolved to the given IP.
- *F-3*: Percentage of IPs in current direct inetnum with DNS records.
- *F-4/F-5*: Mean/Maximum number of TLD+3 domains resolved to IPs in current direct inetnum.
- *F-6/F-7*: Mean/Maximum number of TLD+2 domains resolved to IPs in current direct inetnum.
- *F-8*: Percentage of IPs in /24 IP prefix with DNS records.
- *F-9/F-10*: Mean/Maximum number of TLD+3 domains resolved to IPs in /24 IP prefix.
- *F-11/F-12*: Mean/Maximum number of TLD+2 domains resolved to IPs in /24 IP prefix.

Features from IP Whois. The rest 23 features are retrieved from IP Whois, in other words, the 24 historical snapshots of IP Whois captured in the last 24 months. Here, historical direct inetnums means the 24 direct inetnums in corresponding 24 historical snapshots while historical direct owners and historical loose owners share similar meanings.

- *F-13*: # of unique historical direct inetnums
- *F-14 to F-18*: Current/Maximum/Mean/Minimum/Standard deviation of the sizes of historical direct inetnums.
- *F-19 to F-23*: Current/Maximum/Mean/Minimum/Standard deviation of the depths of historical direct inetnums.
- *F-24*: # of unique assignment types of historical direct inetnums
- *F-25*: Assignment type of the current direct inetnum

- *F-26*: # of current direct owners
- *F-27*: # of historical direct owners
- *F-28*: the percent of current direct owners over historical direct owners
- *F-29*: # of direct inetnums of the current direct owners
- *F-30*: # of IPs of the current direct owners
- *F-31*: # of current loose owners
- *F-32*: # of historical loose owners
- *F-33*: the percent of current loose owners over historical loose owners
- *F-34*: # of direct inetnums of the current loose owners
- *F-35*: # of IPs of the current loose owners

Figure 9 shows the CDFs for some example features on our labeled training set including 10K residential and 10K non-residential IPs.

Evaluation and results. Using the training data of 10K residential IPs and 10K non-residential IPs, we train classifiers of three types: Support Vector Machine (SVM), Random Forest (RF) and Decision Tree (DT). We further evaluate the effectiveness of the models by 5-fold cross validation, testing them on the rest of the four labeled datasets as well as the unlabeled dataset (the RESIP IP dataset) with sampled manual validation.

• *5-Fold cross validation.* We explored the three classifiers with various parameters. 5-fold cross validation reveals random forest with 50 trees outperforms others, achieving the precision of 95.61% and the recall of 97.12%.

• *Testing on the labeled set.* We test the random forest model on all ground truth sets shown in Table II (only those not selected for training). As shown in Table XII, overall the classifier works well. However, surprisingly, it detects 2.45% of IPs in Alexa top 1M set as residential IPs. We find that the domains of those IPs often belong to small local organizations (e.g., local governments or small education institutions) who access the network through residential ISP networks. Another interesting finding is that 65.81% of public proxies (most are either Tor relays or proxy IPs from KuaiDaili service) are predicted as residential, indicating Tor network's effective recruitment of relay volunteers, and also the suspicious proxy sources of KuaiDaili service.

• *Manually validating on the unlabeled set.* We also apply the random forest model on 6.2M RESIP IPs we collected (see §III-A). We detect 5.9M (95.22%) residential IPs and 0.3M (4.78%) non-residential IPs. To evaluate the results, we randomly sampled and manually validated 1K RESIP IPs. Our validation was based upon a set of indicators identified manually. In particular, we searched the Internet to find out whether the owner of a given IP, as indicated in its Whois record, is an ISP or an organization; further we searched the IP itself, which if utilized for a hosting service, most likely was analyzed and reported by the IP information websites such as http://whatismyipaddress.com/ip. The reason we used those as indicators instead of classification features for manual validation is the former are easier for human to tell. Also some of the services have rate limits, prohibiting large-scale automated queries. Our validation shows that the classifier achieved a high precision, 95.80%.

Dataset	Label	% resi	% non-resi
Device Search Engines	resi-clean	98.47%	1.53%
Trace My IP	resi-noisy	94.36%	5.64%
Filtered IP Whois	resi-noisy	99.10%	0.90%
IoT Botnets	resi-noisy	98.82%	1.18%
Public Clouds	non-resi-clean	0.39%	99.61%
Alexa Top 1M	non-resi-clean	**2.45%**	97.55%
Public Proxies	non-resi noisy	**63.54%**	36.46%
RESIP IPs	Unknown	95.22%	4.78%

TABLE XII: Evaluation results of our residential classifier on various datasets. Last two columns show the percentage of IPs in the given dataset being predicted as residential or non-residential.

MD5	Name	Providers
74ac25ba1fa653041b3e2a3d60ceb1d0	hola_svc.exe	LU, IAPS
707ffb5567bf730136614d3356a7d3c5	csrss.exe	PR
7971ebdb5da5c60d0b3f3d8523d94ec7	svchostwork.exe	GS, PO
6925e54c4aecd522230f5765aa6e5a29	swufeb17.exe	PO
2639cd8da42d90a2e112c3d7d3e35540	netmedia.exe	GS, PO
7b024bb2efa5428bbd04f513849cc185	start.vbs	PO
e7dca36767fadfded989ed67e23c2eda	cloudnet.exe	PR
b4b595be616779d4a557cdb49b1350d0	hola_plugin.exe	LU
d85dab7b7112af3feda144bbbffa9b49	produpd.exe	PR
c0a3b6dbbb454a7f3f345d7a87f8e487	pprx.exe	PO

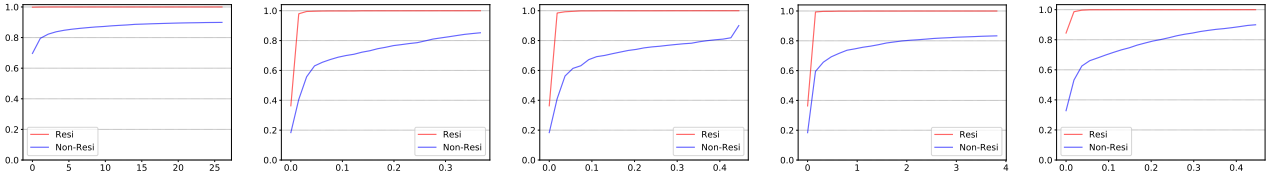
TABLE XIII: List of the top 10 PUPs with their MD5.

B. Botnet Connections

We studied whether IoT botnets are involved in RESIP services. Through cross-matching our RESIP IP database with two botnet IP blacklists (Hajime [12] and IoT Reaper [13], see §III-D), we found 1,248 IPs reported by at least one blacklist on the same way when serving as RESIPs. We further discovered 28,097 RESIP IPs blacklisted between July 2017 and Nov 2017. These findings indicate that at least some resources are shared between RESIP services and botnets, due to either co-hosting of both bots and RESIP software on the same residential system or co-existence of the RESIP system and the bot-infected system behind the same NAT.

C. Others

Datasets and Code release. We will continue collecting and profiling more RESIP services and their RESIPs. Using the techniques developed in this paper, we are working on publishing a service at http://rpaas.site where users can query using a network prefix and obtain a comprehensive report on how the prefix has been used as RESIPs. We will also release weekly snapshots of our RESIP dataset, groundtruth datasets for our residential IP classifier, and all source code of this work once this paper is published.



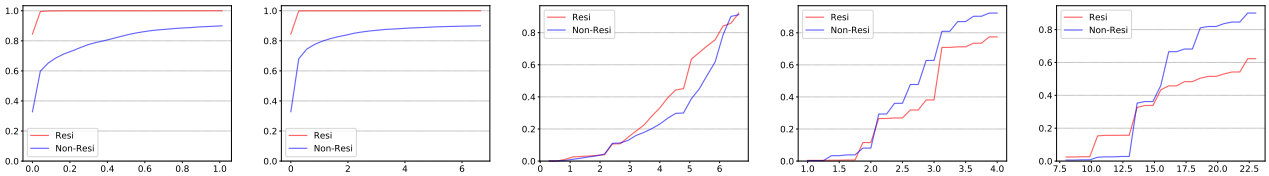
(a) F-2: # of TLD+3 domains resolved to the given IP.

(b) F-3: Percentage of IPs in current direct inetnum with DNS records.

(c) F-4: Mean number of TLD+3 domains resolved to IPs in current direct inetnum.

(d) F-6: Mean number of TLD+3 domains resolved to IPs in current direct inetnum.

(e) F-8: Percentage of IPs in /24 IP prefix with DNS records.



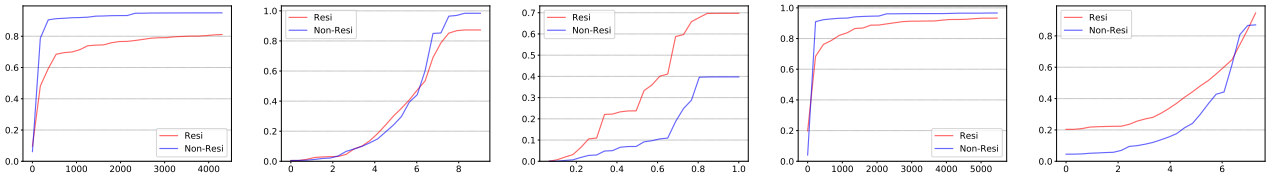
(f) F-9: Mean number of TLD+3 domains resolved to IPs in /24 IP prefix.

(g) F-11: Mean number of TLD+3 domains resolved to IPs in current direct inetnum.

(h) F-17: log_{10} Mean value of the sizes of historical direct inetnums

(i) F-21: Mean value of the depths of historical direct inetnums

(j) F-25: Assignment type of the current direct inetnum



(k) F-29: # of direct inetnums of the current direct owners

(l) F-30: log_{10} # of IPs of the current direct owners

(m) F-33: the percent of current loose owners over historical loose owners

(n) F-34: # of direct inetnums of the current loose owners

(o) F-35: log_{10} # of IPs of the current loose owners

Fig. 9: Cumulative distribution functions of example features on our labeled training dataset.