Geolocalization of Proxied Services and its Application to Fast-Flux Hidden Servers

Claude Castelluccia, Mohamed Ali Kaafar, Pere Manils, Daniele Perito
INRIA Rhone-Alpes
Grenoble – France
{ccastel, kaafar, manils, perito}@inrialpes.fr

ABSTRACT

Fast-flux is a redirection technique used by cyber-criminals to hide the actual location of malicious servers. Its purpose is to evade identification and prevent or, at least delay, the shutdown of these illegal servers by law enforcement.

This paper proposes a framework to geolocalize fast-flux servers, that is, to determine the physical location of the fast-flux networks roots (mothership servers) based on network measurements. We performed an extensive set of measurements on PlanetLab in order to validate and evaluate the performance of our method in a controlled environment. These experimentations showed that, with our framework, fast-flux servers can be localized with similar mean distance errors than non-hidden servers, i.e. approximately 100 km. In the light of these very promising results, we also applied our scheme to several active fast-flux servers and estimated their geographic locations, providing then statistics on the locations of "in the wild" fast-flux services.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Client/Server

General Terms

Measurement, Experimentation, Security

Keywords

Geolocalization, Fast-Flux, Hidden Servers

1. INTRODUCTION

Cyber-crime is consolidating as a major threat for end users and infrastructures on the Internet. Criminals are employing ever changing and more sophisticated techniques to improve the effectiveness, reliability and stealthiness of their illegal activities. Entire underground infrastructures of compromised computers, called botnets [11], have been created to perform a wide range of illegal activities like sending unsolicited e-mail messages, identity theft, disrupting the availability of online services, etc.

IMC'09, November 4–6, 2009, Chicago, Illinois, USA.

An emerging new use of botnets by cyber-criminals is a technique called fast-flux [8], which allows them to reliably host illegal content within a botnet. The goal of this technique is to associate to a fully qualified domain name (such as www.malicious.com) multiple IP addresses that change rapidly and constantly. These IP addresses are chosen in a round-robin fashion from a pool of thousands addresses of the infected machines in the botnet [8]. DNS responses are set with very short Time-to-Live (TTL) to constantly change the resource records that are returned when resolving a fast-flux domain. These IP addresses belong to infected computers that do not host the content but are used as proxies to redirect requests to the actual hidden server.

The study of these malicious networks by security researchers is made difficult by the use of encrypted and obfuscated communications between the participating nodes inside botnets. This calls for research in non-invasive network measurement techniques on botnets in order to understand the way they are used, possibly uncovering the motivations behind them.

Recent promising proposals [4, 9, 2] within the network community, provide ways to reveal either geographic coordinates or network virtual coordinates of Internet hosts. The general idea used to geolocalize Internet hosts is to consider a set of landmarks measuring network distances towards targets and then consider a model that represents a relationship between the geographic distance and the network measurements. Such model, once calibrated, is used by each landmark to derive a geographic distance towards the target, that is then located using multilateration.

In this paper, we extend the *Constraint-Based Geolocation* (CBG) technique [4] to proxied communications, revealing in particular the geographic position of the roots of fast-flux networks. We perform an experimental evaluation of the accuracy of localization in a controlled environment, using the PlanetLab infrastructure, where the exact location of targets is known. Our experimentations show promising results, with geolocalization accuracy similar or even better than non-proxied communication. In particular, we are able to localize hidden servers with mean error distance below 100 km. A vast majority of the obtained confidence zones, a zone where the target lies with a very high probability, allows for a resolution at the regional or even city level, similarly to the original non-proxied system.

In the light of these encouraging results, we tested our geolocalization approach in the wild and located several fast-flux servers. We then validated these results by infiltrating the Waledac fast-flux network to retrieve the IP addresses of some malicious servers.

Finally, since we believe that our study is a first step towards active countermeasures against malicious hidden servers, we provide both our controlled measurements and experimentations on PlanetLab along with traces that we collected from real life experiments

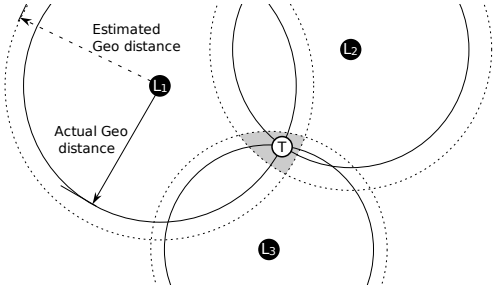

Figure 1: Multilateration with geographic distance constraints. The overestimation of the real distance leads to the creation of the confidence zone in which the target host T will be found.

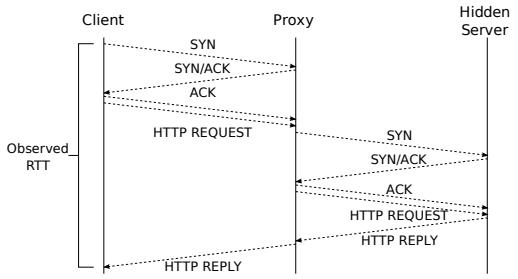

Figure 2: Typical messages exchange in a proxied network. The Observed RTT is approximately equal to twice the RTT between the client and the hidden server. This message exchange has also been observed in our honeypot setting.

on the Waledac Botnet. This data is available for download from `planete.inrialpes.fr/projects/geoloc`.

2. GEOLOCALIZATION TECHNIQUES

Several techniques have been recently proposed for locating Internet hosts [4, 1, 2, 9]. Apart from network positioning techniques, aiming at computing relative network distances between nodes, recent research has also focused on revealing geographic location of Internet hosts based on the network characteristics they exhibit. In this section, we concentrate on one of the most popular techniques, namely Constraint-Based Geolocation [4], proposed to geolocate Internet hosts. CBG is a delay measurement technique that exploits correlations that exist between network distance and geographic distance. It is based on two main phases: the *calibration* and *multilateration* phases.

Calibration phase. This stage consists in estimating the correlation between network and geographic distances in the network. A set of controlled landmarks, whose geographic location is known, probe each other, so that the known geographic distance and the observed network distances can be used to derive the correlation parameters. The model used in CBG considers the correlation between network distance, typically round trip times denoted by (RTT), and geographic distances D as linearly dependent. More formally, for each landmark L_i, its network distance towards node j can be expressed as:

$$RTT(L_i, j) = s_i \times D(L_i, j) + b_i \quad (1)$$

where s_i denotes the slope of the linear model as observed by the landmark, whereas b_i is its interceptor and j is the target towards which the landmark is measuring. The model calibration consists in landmarks pinging each others to collect enough measurements to retrieve an estimation of the values of both s_i and b_i.

Network distances can be influenced by many factors, such as triangle inequality violations and queuing delays, which in turn affect the estimated geographic distances. In order to deal with such factors, each landmark computes a so called best line, as the line that is closest to, but below, all data points (D, RTT). Put simply, each landmark's best line could be considered as the linear model that captures the correlation between D and a RTT that has been the least impacted by the varying network conditions. The distance that separates other data points from the best line, corresponds then to a confidence distance (i.e. an overestimation) that is introduced by the CBG approach to take into account different factors that can impact network distance measurement.

Multilateration Phase. In the second step, given the geographical location of the landmarks and their estimated geographical distances to a given target host (inferred from the model), an estimation of the location of the target host is achieved using *multilateration*. Using the best line computed in the first step, each landmark converts a measured network distance $RTT(L_i, j)$ towards the target j, into a geographic distance $D(L_i, j) = \frac{RTT(L_i,j)-b_i}{s_i}$. In order to localize the target, landmarks cooperate by providing each of their estimated distances towards the target, so that the location estimate is composed by the intersection of the areas provided by the landmarks estimate of the target position (as illustrated by Fig. 1). The target lies somewhere within that area with a very high probability. Interested reader should refer to [4] for further details.

3. GEOLOCALIZATION APPLIED TO PROXIED NETWORKS

The previous section has shown how a target can be geolocalized using a set of landmarks that measure the network distances between themselves and the target. It has also been shown in [4] that this technique is powerful and robust when considering a moderate number of landmarks, ranging from 70 to 100 landmarks.

However, this approach is only effective to locate public servers, i.e. whose IP addresses are known. More specifically, it is assumed that (1) the server responds to ping messages and (2) the pings follow the shortest paths from the landmark to the server. These two assumptions do not hold in the case of proxied servers, such as fast-flux ones. In fact, fast-flux servers cannot be probed with pings, because their IP addresses are not known, and are only accessible through proxies. This also means that, in general, messages from the landmarks to the hidden servers do not follow the shortest paths.

The rest of this section describes two extensions to the CBG scheme to allow geolocalization in proxied-networks. The first extension is used by the landmark to evaluate the RTT to the server using HTTP messages. The second extension is used by each landmark to obtain the shortest RTT to the hidden server in order to reduce its distance estimation error.

Estimating the RTT using HTTP messages. From each landmark point of view, the hidden server, denoted HS, can be seen as an one-hop away node. Its network distance to the hidden server could be measured as a function of its network distance to the proxy and the network distance between the proxy and the hidden server.

Figure 2 shows a diagram of the protocol exchange between a client and a proxied server. When a client, in our case a landmark L_i, wants to request a page from the hidden server HS, it first connects to the proxy, initiating a TCP connection and sending a

HTTP request. The proxy then establishes a TCP connection with the hidden server and relays the HTTP request. The hidden server generates the reply and sends it to the proxy that relays it back to the client. This observation is key to correctly estimate the RTT of data packets between the client and the hidden server. If we denote by *HTTP_Ping* the amount of time elapsed from the first TCP connection to the time the HTTP reply is received by the client, then the RTT can be derived up to a factor, that we call *RTT_factor*.

Such factor is likely to be close to 2, because of the symmetry of exchanged messages as shown in Figure 2. However, since queueing and processing delays can impact RTT_factor, and because the $HTTP\_ping$ is an application-layer time estimation, we need to validate such factor throughout measurements. The estimation of the RTT_factor is reported in Section 4.

Estimating the shortest path. Once the landmark has computed the RTT to a server it can estimate its distance using the model calibrated during the *calibration phase*.

However, hidden servers are accessible through proxies which introduce a level of indirection and therefore, increase the computed RTTs (and resulting distances). In fast-flux networks, to our advantage, the used proxies change very frequently and are distributed over the Internet.

A landmark can then compute the $HTTP\_Ping$ to the hidden server via several proxies and uses the smallest value as an estimation of the $HTTP\_Ping$ via the shortest path. In fast-flux networks, hundreds of proxies can be discovered through time, providing a very good estimation of the $HTTP\_Ping$ on the shortest path.

Once the minimum $HTTP\_Ping$ is obtained, an approximation of the RTT, denoted by $\widehat{RTT}(L_i, HS)$ is computed as follows:

$$\widehat{RTT}(L_i, HS) = min_{p \in \mathcal{P}} \frac{(HTTP\_Ping(L_i, p, HS))}{RTT\_factor}$$

where $\mathcal{P}$ is the set of proxies that the landmark is aware of, and $HTTP\_Ping(L_i, p, HS)$ is the time spent from the TCP connection initiation to the reception of the first HTTP packet from the hidden server, through the proxy p.

Note that another approach consists in mimicking the behavior of data traffic as observed in communication within a fast-flux network, to calibrate the model taking into account the proxying operations. From this perspective, when performing inter-landmarks measurements to evaluate each best line, landmarks emulate a fast-flux network. The idea behind this strategy is that if the network-geographic distance model, and especially its associated best line within each landmark, is calibrated with a fast-flux network-like system, then translation from proxied measurement towards geographic distance can be achieved directly [1].

4. VALIDATION ON PLANETLAB

The goal of this section is to assess the performance of the proposed approach to localize proxied servers. All results were acquired using nodes deployed on the PlanetLab infrastructure, a controlled environment, where the geographic coordinates of each node are provided. The results of these experiments were then used to perform geolocalization of real fast-flux domains.

[1]We have evaluated the performance of this method experimentally. However, since our approach provided better performance, the results of this alternative technique are not reported in this paper.

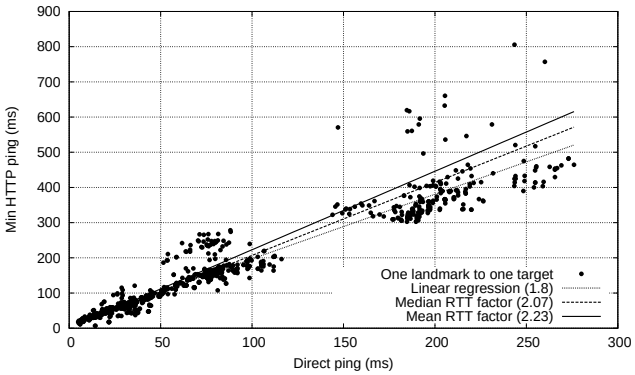

Figure 3: The relationship between Min $HTTP\_ping$ and direct pings towards targets.

4.1 Experimental methodology

For comparison purposes with non proxied communication, we ran our experiments on datasets that are similar to those used in [4]. We performed our experiments using two different data sets with hosts that are geographically distributed through the continental U.S (25 nodes) and Western Europe (30 nodes). Since the location of each PlanetLab node is known, we can evaluate the performance of our scheme by comparing the estimated position to the actual one.

In our experiments, each host plays, one at a time, the role of "hidden" target to be located. The remaining nodes, are then considered as either landmarks or proxies through which communication is achieved to the target. More specifically, the geolocalization of the target node T is performed using the three following steps:

- *Calibration step*: each node (or landmark) computes its best line by pinging each other node (except the target one T).

- *Distance evaluation step*: each node sends HTTP requests to the target node via the other nodes (excluding the target node) and records the corresponding *HTTP_Ping*. It then selects the smallest *HTTP_Ping* value and estimates its distance to the target using its best line.

- *Multilateration step*: all the estimated distances of each landmark are used to compute the estimated location of T as presented in Section 3.

All our experiments were run concurrently so as to experience the same network conditions. Our PlanetLab measurements campaign were conducted between April 15th, 2009 and April 20th, 2009. Since we observed similar results for U.S and Western Europe datasets, in this section we only show U.S results. Moreover, we compared our results to the geolocalization of non-proxied targets.

4.2 Calibration Considerations

In Section 3 we made the assumption that network distances between landmarks and hidden servers can be derived from the $HTTP\_Ping$ measurements and a factor RTT_factor. In an idealized case of proxied communication, and in particular in the case of fast-flux networks, we observed from figure 2, that such a factor can be approximated to 2. However, in practice it is different due to network or queuing delays, application-layer overhead, etc., and must be evaluated experimentally. Figure 3 plots the minimum $HTTP\_ping$ (that each landmark has observed towards a proxied

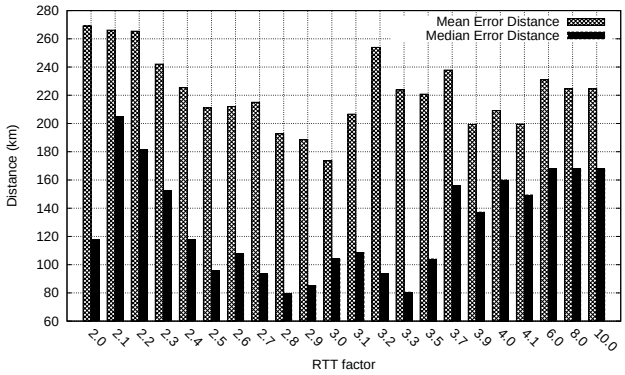

Figure 4: ED. Mean and median error distances in function of the factor RTT_factor.

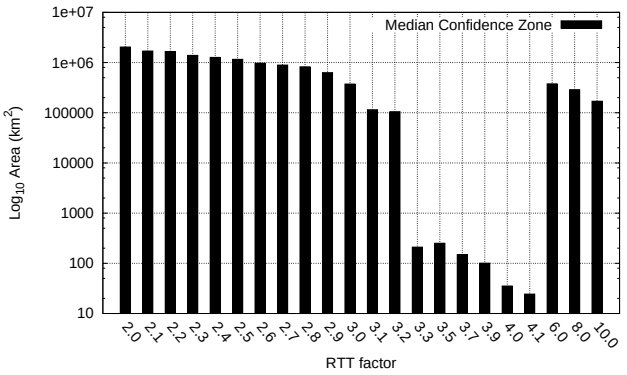

Figure 5: CZ. Median Confidence Zone variation according to the factor RTT_factor. Logarithmic scale on y-axis.

target) against the direct network distance (measured as an RTT), as observed during the PlanetLab experiment. A global first view of the plot shows that a linear relationship effectively exist between the two variables. The ratio between the Minimum $HTTP\_ping$ and the direct ping values gives us an indication about the actual RTT_factor. The median and mean values are respectively 2.07 and 2.23. We depict those RTT_factor estimations using a line $y = RTT\_factor \cdot x$. We also computed the linear regression [3] between the two variables, as an indication of the best linear fitting that could be considered.

It should however be noted that as mentioned in section 3, the number of proxies impacts the trend of the ratio. The more proxies we request to relay the HTTP Pings, the closer the RTT_factor to 2. This is explained by the convergence of the minimum HTTP Pings towards the shortest direct path[2].

We observe that although the actual ratios are close to the idealized case of an RTT_factor equal to 2, proxied $HTTP\_ping$ are often impacted by different network conditions and application layer factors, so that ratios are higher than 2. We therefore computed the location estimations using several values of this RTT factor. The results are reported in the rest of this section.

4.3 Analysis of the Results

This section evaluates the performance of our localization scheme, for different RTT_factor values, according to the three following parameters:

1. *Error Distance (ED):* is the distance between the estimated location and the actual location of the target.

2. *Confidence Zone (CZ):* is the location area identified by our scheme, i.e. the possible geographical area where the target is located. The smaller the area, the more accurate the result is.

3. *Location Error Probability (LEP):* is the probability that the target is not in the confidence zone defined by the scheme. In some cases, the algorithm returns a confidence zone that is incorrect, i.e. that does not contain the actual location of the target. The lower the LEP is, the more reliable the scheme is.

Figure 4 and Figure 5 display respectively the error distances and confidence zones for difference values of RTT_factor.

[2]Triangle Inequality Violations that occur in the Internet [13] may disturb the shortest path routing, we then observe few points where Min HTTP pings are lower than ICMP pings.

These figures clearly show that an RTT_factor in the range 3.3 and 4.0 provides the best localization performance. These values achieve a median distance error of approximately 100 km and a median confidence zone of 100 km^2, which allows for regional and even city-based localization of malicious targets.

The CDF of confidence zone areas (not shown here due to space constraints) confirm that an RTT_factor of 3.3 is a good compromise, with an assigned confidence region with a total area less than 10^5 km^2 for around 80% of the location estimates. This area is slightly larger than one of the smallest U.S states. A confidence region less than 10^3 (corresponding to the area of a metropolis) is achieved by roughly half of the proxied targets estimations. The confidence zone increases again when using a too large RTT_factor, because in cases of non crossing estimations of the landmarks, CBG estimates the position of the target as the geographic location of the closest landmark (i.e. the landmark that estimated the lowest geographic distance). In such a case, since we do not deploy a large number of landmarks, the confidence area increases as it is computed as the area of the circle that would have the closest landmark as a center.

The cumulative distribution function of the observed error distance, across all localized targets, is shown in figure 6. This figure confirms that, choosing an RTT_factor within the range of $[3, 3.5]$ allows for accurate localization. It is worth noticing that, when using an RTT_factor of 3.3, roughly 90% of the proxied targets were localized with an error less than 400 km. Although other curves show higher error distances, the steeper slope of these CDF, compared to the non-proxied curve, shows that the RTT_factor is indeed impacting the relationship between network and geographic distances that is assumed by the constraint based geolocation approach. This is mainly due to the fact that when increasing the RTT_factor, the distance over-estimation that is considered by each landmark in equation 1 is reduced. In other words, when increasing the RTT_factor, the distance that separates other data points from the best line decreases, and so the confidence distance; Since we estimate the location of the target as the centroid of the area of intersection of landmarks estimate of the target position, and since that area decreases when reducing the overestimation, such choice leads to smaller error distances.

A too large RTT_factor may also lead to either non intersection of the landmarks estimates of the target's position, or to non accurate estimation of the confidence zone, and to a large LEP. However, when varying the RTT_factor, we observed that most of the confidence zones (more than 80%, i.e. a LEP of 20%) actually contain the targets. This shows that our approach ex-

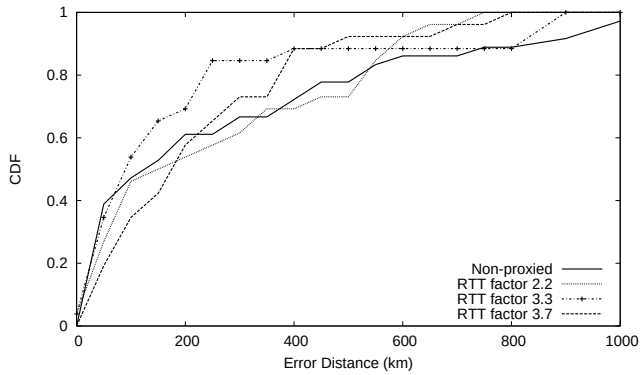

Figure 6: ED. CDF of error distance for different values of RTT_factor.

hibits a high ratio of successful localization inside the confidence zone. Smaller RTT_factor results in a LEP of 10% but provides higher error distances. There is a clear trade-off between the accuracy provided by the scheme and the risk that the reported location is incorrect.

5. LOCALIZING HIDDEN FAST-FLUX SERVERS IN THE WILD

In order to validate our results in real scenario we studied the behaviour of a real fast-flux campaign in the wild. This was achieved by infiltrating the fast-flux network and having one of our computers act as a fast-flux proxy. That in turn allowed us to trace the mothership server IPs and use this information to ultimately validate our previous results.

In our test, we decided to infiltrate a botnet commonly referred to as Waledac [7], a botnet that shares part of its infrastructure with the now defunct Storm botnet [6]. This particular botnet has been chosen because of its sophisticated use of fast-flux inherited from Storm.

Our experimental setup consisted of a single Honeypot [10] equipped with a Virtual Box [12] virtual machine. Five different instances of Windows XP were hosted on this machine, each having its own public and non firewalled IP address. The guest Windows machines were each infected with a copy of Waledac [3]. Special cautionary measures were taken in order to prevent the infected honeypot from participating in harmful or illegal activities, such as blocking outgoing SMTP traffic. The honeypot has been in activity for two weeks at the end of April 2009. A network sniffer was placed on the host machine to monitor and filter all the traffic on the infected machines.

As expected, according to previous research [8], after two days of activity our honeypot has been selected to become a fast-flux proxy, which allowed us to have an in-depth view of the fast-flux hosting infrastructure used.

During the two weeks of activity we were able to log 54 different fast-flux domains and, by looking at the logs, identify the IP addresses of the mothership servers. The domain names were then geolocalized using the technique described in the previous sections. The IP addresses of the motherships were not used as input to our geolocalization tool, but served only to validate our approach. In these experiments 30 PlanetLab nodes were used to perform the measures. An RTT factor of 3.3 was used in accordance to the best results of previous validation on PlanetLab. On average 182 fast-flux proxies were resolved through the DNS and used for each domain.

[3]The malware binary was retrieved on `http://www.offensivecomputing.net` with MD5 hash: $b9ffce1c39cb554510e7c47caec26750$

Domain (.com)	Country	Error Distance	Total Proxies	Avg. Proxies
smsinlinea	PL	173.1	176	46
terrorfear	GE	237.7	189	47
besthandycap	GE	377.2	165	54
orldlovelife	GE	406.8	178	47

Table 1: Localization of few wild fast-flux domains using 30 Europe landmarks.

Table 1 displays a sample of the list of domains that were retrieved using the honeypot and then localized using our technique. An IP-to-geolocation database [5] has been used to obtain the actual position of the mothership server and to compute the error distances. The "total proxies" column displays the total number of unique proxies discovered for each domain. The "avg. proxies" column displays the average number of proxies used by each landmark to geolocalize the target.

Note that, as illustrated by Figure 7, the performance of our scheme increases with the number of proxies. In our scheme, each landmark evaluates the distance of the server by selecting the proxy that provides the smallest $HTTP\_Ping$ value. Therefore, the largest the number of proxies, the higher the probability that the minimum proxied ping is close to the actual direct ping, and therefore the more accurate the localization result is. To our advantage, fast-flux servers use hundreds or thousands of proxies. It should finally be noted that the number of landmarks has an impact on the localization accuracy, as shown in [4].

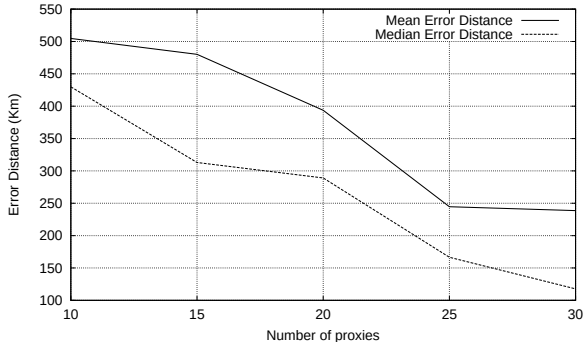

Figure 7: Impact of the number of proxies on the error distances.

The error distances for all the domains lied within a 700 km range, which is in line with our validation on PlanetLab and outperforms the results of the non proxied case. An extensive list of these results can be found on `planete.inrialpes.fr/projects/geoloc`.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a framework to geolocalize proxied hosts in the Internet. We assessed its performance by applying it to malicious hidden servers localization in fast-flux networks.

Despite the possible non optimal deployment of our measurement infrastructure, with a limited number of landmarks, the results obtained show the effectiveness of our method in localizing

proxied servers. Given the gain offered by our approach in terms of accuracy, one can envisage a strategic deployment of landmarks to allow for much more accurate localization of malicious servers, which would considerably help law enforcement.

It might be argued that infiltrating a fast-flux botnet, as we did in Section 5 for validation purpose, is the most efficient and accurate way to localize the hidden servers behind it. We argue that such an approach has several major drawbacks: it requires extensive resources to set up; might incur into legal problems by effectively participating in illegal activities; require knowledge of which specific botnet is using a specific domain, information that is notoriously difficult to obtain. We note that our approach only requires network probes to be sent, and in that, is non intrusive. It can be used as a first monitoring tool in order to assess the possibility of more decisive actions.

Future work will consider possible counter-measures against our localization scheme. In essence, malicious servers, might add random delays in proxies to prevent or disrupt geolocalization. Although, this approach would be not practical since adding delays would degrade the service provided by the malicious servers, we believe that the delay introduced can be filtered out using suitable calibration techniques at the landmarks level. Another objective of our future work is to extend our scheme to identify, in addition to the location of hidden servers, their actual IP addresses. This would be very useful in order to black-list these malicious servers.

7. REFERENCES

[1] BERNARD WONG, IVAN STOYANOV, E. G. S. Octant: A comprehensive framework for the geolocalization of internet hosts. *In Symposium on Networked System Design and Implementation NSDI (2007)*.

[2] DABEK, F., COX, R., KAASHOEK, F., AND MORRIS, R. Vivaldi: A decentralized network coordinate system. *In SIGCOMM (2004), pp. 15-26*.

[3] DRAPER, R, N., AND HARRY, S. Applied Regression Analysis (Wiley Series in Probability and Statistics), *ed. John Wiley & Sons Inc*.

[4] GUEYE, B., ZIVIANI, A., CROVELLA, M., AND FDIDA, S. Constraint-based geolocation of internet hosts. Networking, *In IEEE/ACM Transactions on 14, 6 (Dec. 2006), 1219-1232*.

[5] HEXASOFT DEVELOPMENT SDN. BHD. ("HDSB"). Ip-to-geolocation db. *http://www.ip2location.com/, 2009 (accessed Apr, 2009)*.

[6] HOLZ, T., STEINER, M., DAHL, F., BIERSACK, E., AND FREILING, F. Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. *In LEETÕ08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (Berkeley, CA, USA, 2008), USENIX Association, pp. 1-9*.

[7] NAZARIO, J. Walking waledac. *http://asert.arbornetworks.com/2009/01/ walking-waledec/, 2009 (accessed Apr, 2009)*.

[8] NAZARIO, J., AND HOLZ, T. As the net churns: Fast-flux botnet observations. *In the 3rd International Conference on Malicious and Unwanted Software, 2008. MALWARE 2008. pp. 24-31*.

[9] NG, E. T. S., AND ZHANG, H. A network positioning system for the internet. *In ATEC Õ04: Proceedings of the annual conference on USENIX Annual Technical Conference (Berkeley, CA, USA, 2004), USENIX Association, p. 11*.

[10] PROVOS, N. A virtual honeypot framework. *In Proceedings of the 13th USENIX Security Symposium (2003), pp. 1-14*.

[11] RAJAB, M. A., ZARFOSS, J., MONROSE, F., AND TERZIS, A. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. *In HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (Berkeley, CA, USA, 2007), USENIX Association*.

[12] SUN MICROSYSTEMS INC. Virtual box. *http://www.virtualbox.org/, 2009 (accessed Apr, 2009)*.

[13] ZHENG, H., LUA, E. K., PIAS, M., AND GRIFFIN, T. G. Internet routing policies and round-trip-times.*In Passive Active Measurement Conference PAM (2005)*.