IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Empirical analysis of Tor Hidden Services

ISSN 1751-8709
Received on 1st April 2015
Revised on 7th September 2015
Accepted on 17th September 2015
doi: 10.1049/iet-ifs.2015.0121
www.ietdl.org

Gareth Owen ✉, *Nick Savage*

School of Computing, University of Portsmouth, Lion Terrace, Portsmouth, Hants PO1 3HE, UK
✉ *E-mail: gareth.owen@port.ac.uk*

Abstract: Tor hidden services allow someone to host a website or other transmission control protocol (TCP) service whilst remaining anonymous to visitors. The collection of all Tor hidden services is often referred to as the 'darknet'. In this study, the authors describe results from what they believe to be the largest study of Tor hidden services to date. By operating a large number of Tor servers for a period of 6 months, the authors were able to capture data from the Tor distributed hash table to collect the list of hidden services, classify their content and count the number of requests. Approximately 80,000 hidden services were observed in total of which around 45,000 are present at any one point in time. Abuse and Botnet C&C servers were the most frequently requested hidden services although there was a diverse range of services on offer.

1 Introduction

The Onion Router (Tor) is a network of routers whose purpose is to anonymise the traffic of a user by mixing traffic with that of others and relaying it through several intermediate hops before forwarding to the destination. Tor is popular amongst individuals concerned with their privacy and is operated by a non-profit organisation which receives funding from a range of sources, paradoxically including the US Government. At the time of writing, Tor has ~2.5 million active users [1] world-wide although this has peaked at almost 6 million in the last 24 months. State-level censorship and surveillance has been a primary driver behind the popularity of Tor [2].

The tor network is a set of Onion Routers (ORs), N, where at the time of writing (October 2014), $|N| = 8000$ [1]. A set of relays $A \subset N$ are denoted authority nodes which are trusted relays and assign flags to nodes based upon their observed behaviour. The set of nodes with assigned flags are called the *consensus,* here denoted as $C \subset N$. Each node must maintain a copy of C and discard and renew it once its expiry time has passed. The authorities designate each OR as having a set of flags which describe certain characteristics, for example, a node which has the exit flag permits a user to exit the Tor network and connect to regular Internet TCP services while a node with the guard flag is eligible to be the first node in a circuit.

A client will use at most three relays to form a route $\{r_{\text{guard}}, r_{\text{mid}}, r_{\text{exit}}\}$, where $r_i \in C$ which can be used to route traffic through (a route is termed as a circuit). Crucially, r_{guard} will be chosen from the set N_{guard} and r_{exit} will be chosen from N_{exit}, where N_{exit} denotes the set of nodes in N with the exit flag. This ensures that strict controls are maintained on those nodes allowed to function as the first or last hop in a circuit (specifically, to increase the cost of operating a node in the set N_{guard}).

A user does not select nodes to use in its circuit at random from the consensus because this can lead to poor performance. Each node in N publishes its bandwidth $n_{i,\text{bw}}$ and this is independently verified by other nodes trusted by more than half of A. A user selects any one node $r_{\text{guard}} \in N_{\text{guard}}$ with probability defined in (1). The same process is used to select an exit. A user will maintain its guard node for a long period of time to reduce the likelihood of an attacker controlling it. If the guard was renewed frequently, then an attacker would be able to control the guard node *at some point* simply by running a large number of ORs [3]

$$\left.\frac{r_{\text{guard,bw}}}{\sum_{i=1}^{|N_{\text{guard}}|} n_{i,\text{bw}}}\right| n_i \in N_{\text{guard}} \tag{1}$$

For every OR $n \in N$, several clients may be using it as a hop in their circuit hence achieving a traffic mixing effect that makes it difficult to trace individual packets through the network. A strong adversary can observe traffic entering and leaving the Tor network and perform a correlation attack to negate the mixing effect on traffic and hence link outgoing traffic to an individual user. In the worst case, an attacker controlling the first and last hop can undertake packet correlation and deanonymise a user with non-negligible probability [4].

The Tor network as described thus far allows a user to build circuits through the network and connect to an ordinary Internet TCP service. The user is anonymous to the service because the service only sees the Internet protocol (IP) address of the last hop in the circuit (the exit node); however, the user is perfectly aware of the identity of the Internet service because its hostname or IP address were required to connect. If an activist in a repressive regime wishes to run a pro-democracy blog without risk of imprisonment, then Tor offers a further facility called hidden services (HS) which permits the web-server to its visitors (e.g. neither party knows the identity of the other). The set of HSs is often referred to as the 'darknet'.

A TCP service (such as a website) that wishes to advertise itself anonymously chooses three nodes from the consensus to use as introduction points (IPTs) and builds circuits to them. The service then publishes a descriptor in a distributed hash table (DHT) to advertise itself and its IPTs. Users wishing to visit the site, fetch the descriptor from the DHT, build a circuit to one of the IPTs and instruct it to forward a message to the HS requesting it to build a circuit to a rendezvous point (RP). The RP is another node in the consensus chosen by the user, and both the user and the hidden site build circuits to it and ask it to relay packets between them; therefore, because both the user and the site have built three-hop circuits to the RP, an attacker gains no advantage by controlling it.

In this paper, we describe what we believe to be the largest study to date on Tor HSs. For approximately 6 months, a large number of

nodes were operated to participate in the DHT (without disrupting service). Publications of and requests for HS descriptors (HSDescs) were collected and then used to seed a web-crawler which then collected a range of data points from each service.

Section 2 discusses existing published work that has examined the Tor network to determine its usage. Section 3 discusses the methodology used to collect the results which are presented in Section 4.

2 Related work

There have been previous attempts to study what Tor is used for although intercepting traffic exiting the Tor network is likely to be unlawful in most jurisdictions. In 2010, a study [5, 6] examined traffic port destinations of traffic leaving an exit node onto the open Internet. At the time, file sharing appeared to dominate with only 1.5% of traffic being to HTTP services. Measuring traffic can easily be skewed by high traffic services such as file sharing as illustrated by a slightly earlier study which identified HTTP traffic as being 92% of connections leaving an exit [7]. Another study specifically examined malicious traffic leaving a Tor exit node by using Snort filters and found that exiting traffic was frequently P2P, botnet, spam or another category of malicious [8].

HSs were described in the original Tor paper [9] and have undergone several revisions since; however, whilst HSs are difficult to locate they use a DHT similar to chord [10] to publish descriptors describing how to connect to them. The Tor DHT is not resistant to Sybil attacks [11], that is, one can run many nodes and gain control of a large proportion of the DHT. From then, he can collect HSDescs publications and more worryingly deny service to legitimate users. There exist many Sybil-resistant DHT proposals (e.g. [12, 13]) but as of yet Tor has not explored these approaches.

At present, the Tor DHT consists of ~2800 ORs with the hidden service directory (HSDir) flag. An OR must be operating for more than 25 h before it may obtain the HSDir flag and one may only operate two ORs per IP address, which makes infiltrating the DHT to collect information a potentially expensive process. One author [14] describes a bug in the Tor core program that allows one to launch a large number of ORs on a single IP address and selectively phase any two into the consensus. Tor logs ORs' uptime even if they were not in the consensus, and so it was possible to launch a number of ORs on single IP address, and after 25 h selectively make some unreachable causing others to enter the consensus. By doing this, the authors were able to collect the list of HS addresses in fewer than 2 days; however, the Tor project has now fixed this bug by only logging uptime for nodes *in* the consensus. The authors used an automated classification algorithm to classify hidden sites which lacked precision and their study only examined HSs present during that 24 h period. Therefore, the general question of the size, content and popularity of the darknet remains an open question which we address in this paper by collecting data over a significantly longer period of time and manually classifying sites to achieve greater precision.

Other work which has examined HSs are those which have inspected the pages published on these sites and attempted to determine popularity information. For example, [15] examined silk road through crawling the listings of products and found that most items for sale were related to drugs.

3 Methodology

Each HS uses a DHT over a subset of ORs to store HSDescs, a text document that describes its IPTs. As mentioned above, only ORs that offer more than 50 kB/s of bandwidth and are active continuously for 25 h or more are eligible to obtain the HSDir flag which allows them to participate in the DHT. These restrictions aim to increase the cost to those wishing to collect data from or manipulate the DHT – although at today's server costs the obstacle is minimal.

The Tor DHT is similar in design to the chord DHT [10] in that DHT participant nodes are mapped onto a circle along with data for storage by use of a hash function. In the case of Tor, its hash function $H: X \rightarrow \{0, 1\}^{160}$ is the SHA-1 pseudo-random one-way function mapping the input set X to the set of bit strings of length 160. The use of a hash function exhibiting strong pseudo-random characteristics is important to ensure even distribution around the circle. Each OR is mapped onto the circle by using $P = (\mathrm{PK}_{\mathrm{OR}})$, where $\mathrm{PK}_{\mathrm{OR}}$ is the ASN.1 encoding of the OR's public key.

Each of the Tor HSs is mapped onto the circle by the use of a unique descriptor ID as defined in (2), where $P = H(\mathrm{PK}_{\mathrm{onion}})$ and $P[a{:}b]$ denotes bytes a through $b-1$ of P, d is an optional descriptor cookie (shared secret) used to provide client-side authentication for HSs which are not accessible to all users. Finally, $r \in \{0, 1\}^8$ is defined as the replica value and may be 0 or 1. The replica value provides a degree of redundancy, by hashing first with a value of 0, and then again with a value of 1, it gives two distinct (with high probability) locations on the DHT for publication of the descriptor ($\|$ denotes concatenation of bit strings)

$$\mathrm{descid} = H(P[0{:}10] \| H[t_{\mathrm{p}} \| d \| r]) \tag{2}$$

The time period t_{p} is defined in (3), given the time t in UNIX time (seconds since 00:00 on 1 January 1970). The effect of this is that t_{p} changes once per day in any one of 256 intervals defined by the first byte of P. This ensures that all HSs do not attempt to change their publication servers at the same time

$$t_{\mathrm{p}} = \left\lfloor \frac{t + P[0{:}1] \cdot (86,400/256)}{86,400} \right\rfloor \tag{3}$$

Each HS, after mapping all ORs and itself onto the DHT circle, publishes its descriptor to the three ORs to the right of its descriptor-id on the DHT. As the HS is mapped onto the circle at two locations, a total of six ORs receive a copy of the HSDesc. The HS publishes the text document by building a circuit to each of the designated ORs and establishing a HTTP connection to its directory port.

Fig. 1 illustrates this, where $h \in N_{\mathrm{HSDir}}$ and h_i denotes an OR with an identity hash following the desc ID with replica = 0, and h_k denotes the same but with replica = 1.

Tor is an open-source project so we were able to download and modify the official client to log all publication requests from HSs along with all requests from clients for a copy of the descriptor. After a period of 25 h, an OR would become eligible to participate in the DHT and hence begins seeing publication requests and requests from clients. As each HS changes the location in the DHT at which it publishes its HSDesc every 24 h, by running just one OR and ensuring it participates in the DHT for a significant period of time, then that HS's descriptor will be collected with high probability (provided the HS also remains online for the duration). We operated 40 servers which significantly speeds

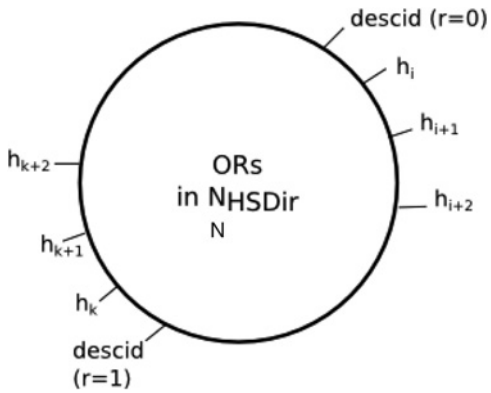

Fig. 1 *Tor DHT*

collection and allows us to observe each HS on many distinct days during the study.

There are two classes of descriptor requests that we may receive: a successful request and a failed request. A request does not contain the HS address but just the desc ID and because this is generated by a non-invertible function one cannot resolve the service address; however, when a HS publishes its descriptor it is possible to link the descriptor ID to the HS address but where we have not seen a publication then the true address is unknown. This naturally occurs because the DHT is constantly changing in size, so there will inevitably be cases where clients request from the wrong OR. Given the learned request time, the desc ID and a large list of HS addresses we had accumulated in our database, we were able to run a brute force attack by repeating the calculation in (2) for each HS address for the given time frame, thereby resolving a large number of failed descriptor requests.

3.1 Understanding the request metric

A user will make a HSDesc request when he first wishes to visit a HS. His Tor client will cache this descriptor until expiry so that repeated attempts to visit the service do not produce a new descriptor request. That said, if the user closes his Tor client between visits then this will force a new descriptor fetch. Furthermore, a descriptor is valid for a fixed period of 24 h, and if the descriptor expires between visits then a new request will be made. Caching HS descriptors to disk is risky because this leaves traces of what the user might have been doing. Therefore, we can say that a HS descriptor request correlates neither exactly with the number of visits or the number of visitors, but is somewhere in between. Precisely where is impossible to measure because we are unable to link individual requests to the same requester.

4 Results

4.1 Darknet size estimation

The number of ORs that are operated in the DHT is directly proportional to the collection *rate* of HS descriptors. The more relays one runs, the larger the proportion of the DHT he observes and hence the faster the collection of HSs.

During the collection period, just short of 80,000 HSs were observed. Fig. 2 shows the cumulative number of unique HS addresses that were observed over the duration, whilst Fig. 3 shows the number of unique HSs observed for each discrete day. Initially, as one might expect, the rate of learning is quickest at the beginning because almost all publications are previously unseen HSs. As time goes on, this learning rate tails off to produce a steady rate of ~2700 new and unique HS addresses per day – which can be said to be the HS birth rate.

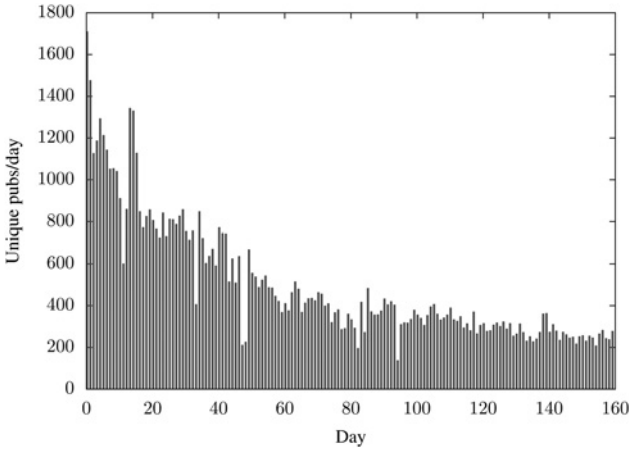

Fig. 3 *Unique HS publications/day*

Estimating exactly the total number of HSs at any one time is impossible without controlling all Tor relays. For each discrete day, we observe a count on total unique HS publications and from this we extrapolate knowing the proportion of the DHT we contain to the estimate for number of HSs (see Fig. 4).

Whilst the data appears very noisy, this is because of the small fraction of the DHT that we are observing. If one takes the long-term average over the entire collection period, the estimated number of HSs is 45,000 – although given the high turnover this could vary significantly from day-to-day.

Fig. 5 shows the number of times a HS published to one of our servers during the study; for example, for over 12,000 onion addresses a publication was only observed on a single day through the duration of the study. This means that those services were very short lived, existing for at most a few weeks before being shut down. Notably, there appeared to be a very high turnover of HSs; that is, many only existed for a short period of time and were never seen again.

Irrespective of the high-turnover rate, there exists a significant number of HSs that are long lived. For these HSs, we would expect to see them repeatedly publish to our servers. Hence, one might therefore expect Fig. 5 to be bimodal, with peaks for short-lived and long-lived services but this is not observed in the data because the long-lived services were dwarfed by the short-lived services. Only 15% of HSs continued to exist for the entire 6-month period.

To further verify the high churn rate, the list of HSs that [14] collected in a 24 h period over 18 months prior were compared

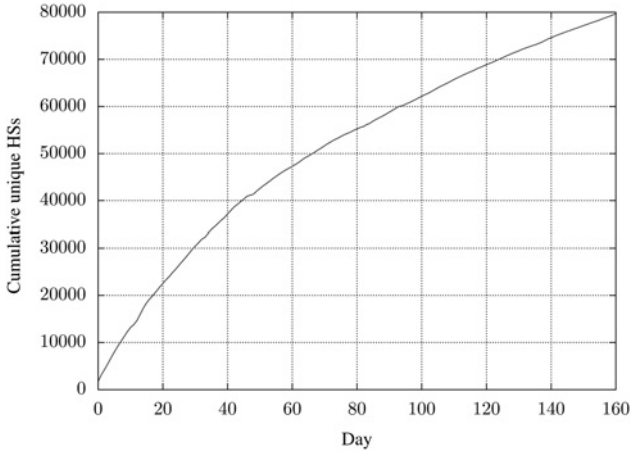

Fig. 2 *Unique HS observed over time*

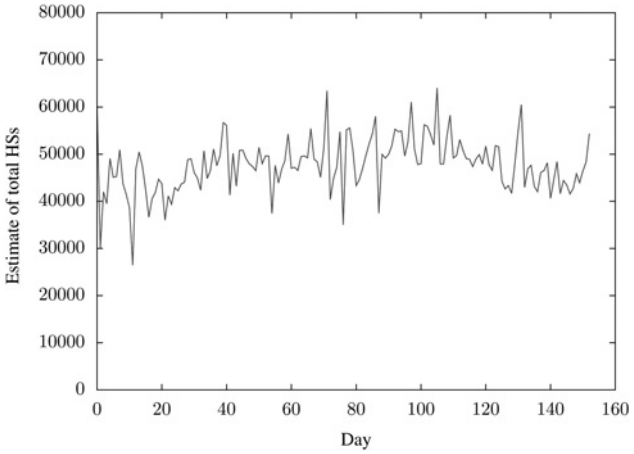

Fig. 4 *Estimate of total HSs*

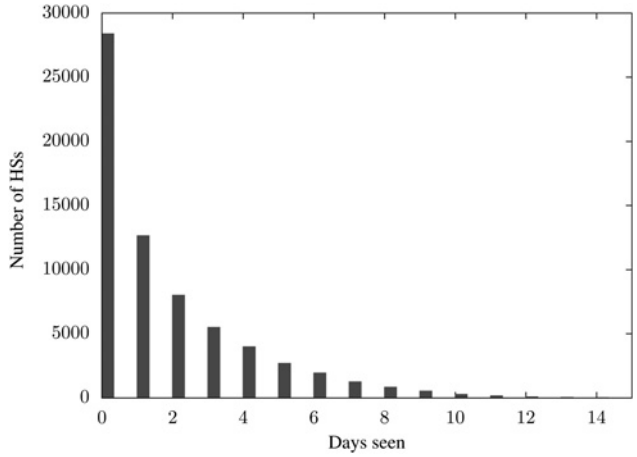

Fig. 5 *Number of days each HS observed*

with this dataset. Fig. 6 shows that only a tiny minority of HSs have persisted for the 18 months, fewer than 10,000 whilst the majority either disappeared or were not seen previously.

4.2 Authenticated HSs

By inserting an additional string into (2), clients can be authenticated to the hidden server. This additional string is used as a password in addition to the onion address so that only users in possession of both can fetch the HSDesc and connect to the HS – thereby authenticating the client using a keyed-MAC function. In addition, the list of IPTs in the HSDesc is also encrypted with this additional string so that no information is leaked to those servers hosting it in the DHT.

We are able to determine whether a HSDesc is for an authenticated HS by inspecting the list of IPTs. For an unauthenticated HS, the list of IPTs will consist entirely of ASCII characters, and be of valid form, whereas an authenticated HS's IPTs will be encrypted with a modern symmetric cipher producing output indistinguishable from random.

We examined every HSDesc and found that ~0.6% were for authenticated HSs while the rest were un-authenticated. Hence, this feature appears to be little used.

4.3 Darknet content

A HS is merely a TCP endpoint with 65,535 available ports. There is no mechanism in Tor to find out which ports are available for use by clients and the only way to discover them is by performing a port-scan. HTTP and HTTPS were nearly universally offered by the popular HSs. To identify the type of content available, a custom crawler was developed that would connect to a local Tor client and establish HTTP/HTTPS connections to the above collected HSs, download HTML content and extract key data points. These data points were then used for classification of the content type.

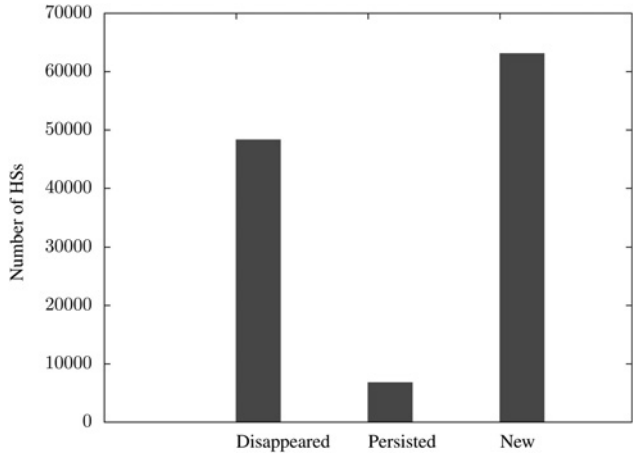

Fig. 6 *Number of HS persisting over 18 months*

4.3.1 Classification: Classification of web pages is a difficult task and whilst there exists automatic classifiers based upon machine learning approaches, the dataset in this case was small enough that manual classification was not onerous. Additionally, the authors felt that given the range and complex technical nature of some of the content that automatic classifiers would be insufficient due to difficulty in interpreting context and meaning [16].

Deciding when to crawl is not straightforward, at first glance one may assume that crawling throughout the study is the logical approach; however, in doing so one will largely collect data about HSs which were short lived most of which were not online concurrently. This is not terribly useful because short-lived services will be over represented. Instead, we preferred to take a snapshot of the content at a particular period in time and having observed the turnover and short longevity of services we crawled over a 1-month period toward the end of the study. This enabled us to capture short-lived services in a way which they are not unduly represented although we acknowledge that without observing the whole of the DHT it is impossible to capture a perfect snapshot and so this represents the best approach we are aware of.

It was considered at the outset that some HSs may contain content which was obscene or otherwise illegal to download and it was more than likely that if this content existed it would be in the form of multimedia or images; hence, our crawler fetched HTML pages from each HS, parsed key data points, stored them in a database – no image, media or javascript were fetched. The crawled data were inspected to produce a list of categories which covered the majority of the content. Afterwards, we manually examined the data points for each HS and classified them into distinct categories. Where a site spanned two categories, the authors chose the category which more precisely described the overwhelming primary purpose.

Whilst there is often debate about the division of illegal and legal content on the darknet, it is difficult to classify sites into either legal or illegal due to discrepancies and intricacies between legal jurisdictions; for example, whistle-blowing sites are often considered legal but may not be if used to disclose classified documents or by persons in repressive regimes. Therefore, a formal classification into legal and illegal has not been undertaken; although we note that the majority of sites appeared to be of questionable morality/legality.

We define those categories which are ambiguous below as follows. *Bitcoin*: currency exchange from a mainstream currency to bitcoin, laundering services and so on. *Drugs*: the sale or purchase of narcotics – typically meaning marketplaces connecting buyers and sellers. *Market*: a marketplace selling items other than drugs or services covered in other categories. *Fraud*: sites attempting to obtain a pecuniary advantage by deception. *Mail*: email or messaging services, including instant messaging and chat. *Whistleblower*: sites typically operated by journalists for whistle-blowers to submit documents. The GlobaLeaks platform [17] and SecureDrop platform [18] were prominently featured in this category. *Counterfeit*: sites offering counterfeit items; notable fake currency such as notes or fake passports/identity documents. *Abuse*: sites where the title indicates some form of sexual abuse (typically minors) likely to be illegal in most western jurisdictions. Sadly, these pages were easily identifiable from the meta data suggesting web-masters had confidence that Tor would provide robust anonymity. For some sites it was difficult to discern whether they were facilitating abuse or providing adult pornographic services, and due to legal restrictions we were unwilling to download images to confirm – where this was the case we preferred to put the site into the *porn* category. *Hosting*: web/server hosting services. *Books*: book sales or electronic pirate copies of ordinary texts.

Table 1 Non-sequential snapshot of well-known valid services

Onion	Requests/day	Days seen	Desc
censored	168,152	12	abuse
silkroad6ownowfk	8067	11	silk road
agorabasakxmewww	3035	8	agora
k5zq47j6wd3wdvjq	2589	5	evolution
xmh57jrzrnw6insl	1341	7	torch
3g2upl4pq6kufc4m	1223	4	DuckDuckGo search
wikitjerrta4ggz4	555	12	HiddenWiki
mail2tor2zyjdctd	266	8	mail

4.3.2 By popularity: Table 1 shows a cross-section of the widely known onion addresses by the number visitors they received each day. The list is not comprehensive for the sake of brevity, as for example, botnet C&Cs alone fill the top 40 spots. A visit was classed as a fetch from the HS directory for the HS's descriptor describing the IPTs. Typically, a Tor client will fetch this descriptor once in any 24 h period and cache it for future access although if the user restarts the tor daemon then the document may be re-fetched. At present, no technique exits to distinguish between a repeat visitor and a unique visit in a 24 h period due to the anonymity offered by Tor.

Amongst valid HSs, abuse sites were by far the most popular sites and as stated earlier these sites were easily identified. The data highlights the shear number of requests going to abuse sites compared with previously thought to be popular sites.

Table 2 shows the popularity of HSs for which we received a descriptor request but never received a publication during the study. These are addresses which no longer exist but are still being requested by Tor clients. In many cases, it was possible to identify the purpose of these now extinct HSs by examining online malware reports or by word prefixes present in the onion address.

Almost all of the top 40 HSs that were invalid are botnet command and control servers [19] which are used to control computers infected with malware (called bots) remotely; the bot will connect to the server regularly for new instructions or to upload data (e.g. stolen passwords). Malware authors and researchers have been involved in a cat-and-mouse game in the recent decade where by authors have attempted to produce C&C servers that are difficult to take down [20]. Tor has become a popular tool for C&C infrastructure for botnets due to the difficulty in taking down and locating servers. Interestingly, most botnets represented in the dataset had many (as opposed to a single) HS addresses which paradoxically may make them more vulnerable to deanonymisation attacks [14, 21–23] if these services are distributed across several Tor processes.

4.3.3 Classification: Two representations of the classifications of data are presented. First, the number of sites in each category is presented in Fig. 7. This figure shows that the content on the darknet is diverse but a large proportion is of questionable legality.

When each category is plotted against the percentage of HS directory requests it received (using the previous hits data), an entirely different picture emerges as shown in Fig. 8. Requests to abuse sites represented over 80% of total requests observed.

Table 2 Popularity of invalid onions

Onion	Requests/day	Days seen	Desc
l77ukkijtdca2tsy	679,470	9	botnet sefnit
7sc6xyn3rrxtknu6	525,930	11	botnet sefnit
pomyeasfnmtn544p	514,766	10	botnet sefnit
—	—	—	sefnit botnet addresses
ceif2rmdoput3wjh	247,296	6	botnet skynet
—	—	—	skynet botnet addresses
censored	6603	10	abuse

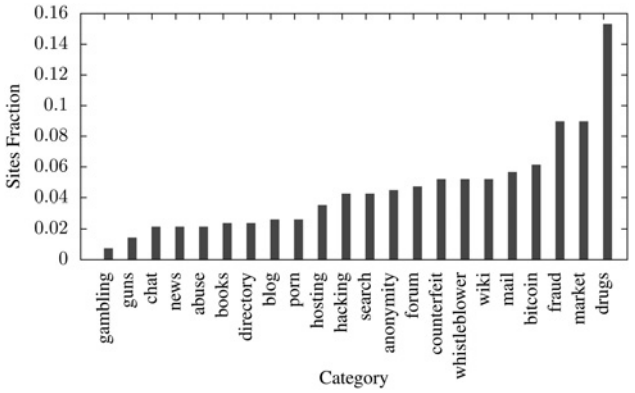

Fig. 7 *Content by number of sites*

4.3.4 Limitations and interpretation: It is important to emphasise what is being measured. The popularity data is a measure of the number of HS directory requests and when grouped into content-type categories the picture may become somewhat misleading. First, law enforcement frequently patrol abuse sites and this may inflate the figures; however, crawlers are likely to account for a single request in a 24 h period and we are seeing a large number of requests to these sites – even if assuming all national forces crawl these sites daily they would still only account for a small proportion of the total requests. The second possibility is denial of service attacks, one could flood the HS directory with requests for descriptors in an effort to take the directory offline – this is likely to be ineffective because the attacker would need to take all six directories offline and then these relays would be dropped from the consensus and the responsibility would shift to other relays. It is worth bearing in mind that most of these sites were observed on several random days during the study so an attack of this nature would have to persist for most of the duration of our study. Whilst it is impossible to rule out due to the anonymity offered by Tor, it seems unlikely and in any case none of our servers were taken offline.

Tor offers a tool called Tor2Web which allows non-Tor users to visit HSs through a web gateway. These web gateways will operate one or a small number of Tor clients and so although there might be several visitors to a site we will only see one request because the gateway has cached the descriptor; hence, it is possible for some sites to be under-represented in the data if they are largely accessed through Tor2Web.

4.4 Darknet server configuration

The crawler also recorded information on the software running on each crawled Tor HS by storing the HTTP server header from the response. As illustrated in Fig. 9, Apache was the most widely used web server software, which is not dissimilar to sites on the

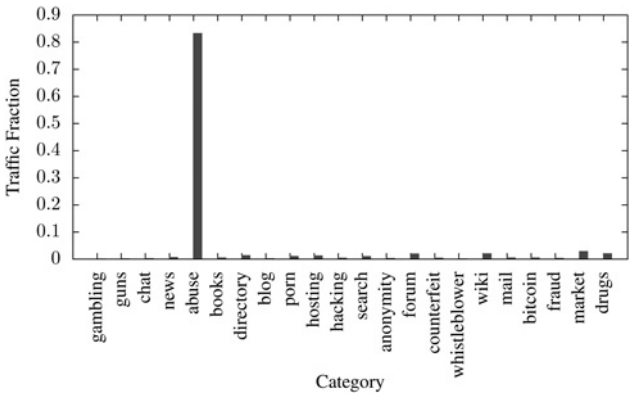

Fig. 8 *Content by popularity*

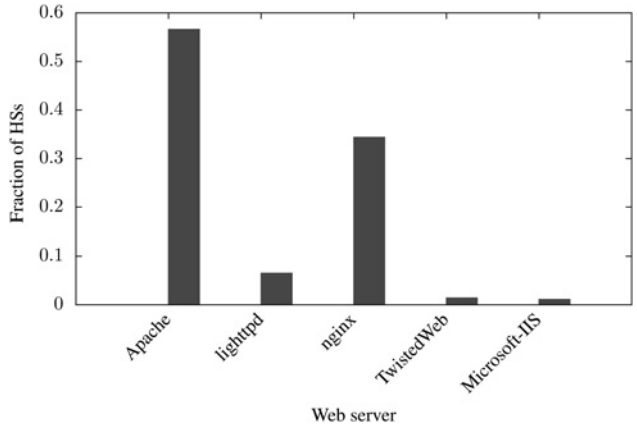

Fig. 9 *Web server software*

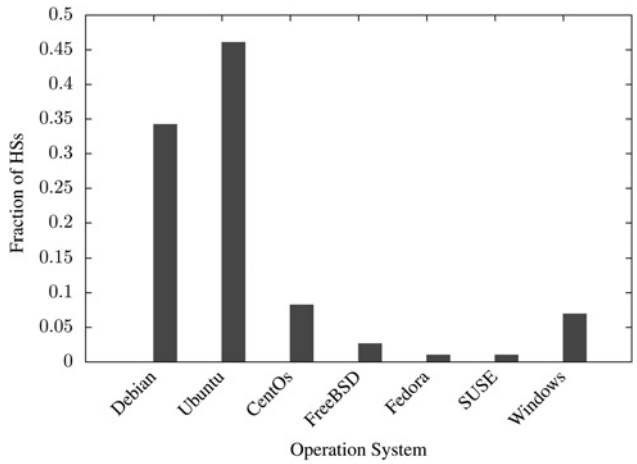

Fig. 10 *Operating systems*

ordinary Internet; however, on the Internet Microsoft's IIS server is also very popular but is drastically under-represented on the servers we observed. Open-source web servers were utilised by almost all HSs except a tiny fraction.

In most cases, the web servers also returned information on the operating system running on the hidden server in the HTTP server header. As illustrated in Fig. 10, open-source software is widely represented, with Debian or the Debian derivative Ubuntu accounting for almost 80% of all servers. Windows servers accounted for <10% whilst the remaining servers were a range of UNIX derivatives (largely Linux distributions).

4.4.1 Darknet connectivity: For each HS website that was crawled, we extracted information on the HTML hyperlinks listed on their site. Each link was categorised into one of three categories: darknet, clearnet or own-site. Darknet links were links to other Tor HSs, clearnet links were links to websites that were hosted on the Internet (e.g. regular non-Tor domains) and own-site were links within the site being crawled.

Of the HSs we crawled, 59% did not link to any site other than itself, 7% linked only to other darknet sites, 23% linked only to clearnet sites and 11% linked to both. Aggregating the first two figures, one can say that two-thirds of sites were not connected by links to any sites outside of the darknet.

5 Conclusion

It is straight forward to collect and monitor Tor HSs without detection by launching a large number of Tor relays provided one keeps below the detection threshold. This places the operator in a powerful position, being able to monitor hits and deny service to legitimate users.

Whilst Tor HSs allow websites and visitors to remain anonymous to each other, supporting a broad range of activities from whistle blowing to gambling, sadly abuse and criminal orientated websites appear to be popular drawing attention away from the legitimate uses.

Future work might examine further the non-HTTP-based HSs although this is expected to be complicated due to the principally manual process and because the Tor protocol now makes port scanning extremely slow. Additionally, services of different types may not be comparable.

6 References

1 Tor Project: 'Metrics portal', September 2014. Available at http://metrics.torproject.org
2 Winter, P., Lindskog, S.: 'How the great firewall of china is blocking tor'. Proc. of USENIX Workshop on Free and Open Communications on the Internet, 2012
3 Elahi, T., Bauer, K., AlSabah, M., *et al.*: 'Changing of the guards: a framework for understanding and improving entry guard selection in tor'. Proc. of the Workshop on Privacy in the Electronic Society, 2012
4 Murdoch, S.J., Zielínski, P.: 'Sampled traffic analysis by Internet-exchange-level adversaries'. Proc. of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007), 2007
5 Loesing, K., Murdoch, S., Dingledine, R.: 'A case study on measuring statistical data in the tor anonymity network', *Financ. Cryptogr. Data Secur.*, 2010, **6054**, pp. 203–215
6 Chaabane, A., Manils, P., Kaafar, M.A.: 'Digging into anonymous traffic: a deep analysis of the tor anonymizing network'. Proc. of the 4th Int. Conf. on Network and System Security (NSS), 2010
7 McCoy, D., Bauer, K., Grunwalk, D., *et al.*: 'Shining light in dark places: understanding the tor network', *Priv. Enhancing Technol.*, 2008, **5134**, pp. 63–76
8 Ling, Z., Luo, J., Wu, K., *et al.*: 'Torward: discovery of malicious traffic over tor'. Proc. of the IEEE Int. Conf. on Computer Communications, 2014
9 Dingledine, R., Mathewson, N., Syverson, P.: 'Tor: the second-generation onion router'. Technical report, Naval Research Laboratory, USA, 2004
10 Stoica, I., Morris, R., Karger, D., *et al.*: 'Chord: a scalable peer-to-peer lookup service for internet applications'. Proc. of the 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2001, 2001, pp. 149–160. Available at http://doi.acm.org/10.1145/383059.383071
11 Douceur, J.R.: 'The sybil attack'. Revised Papers from the First Int. Workshop on Peer-to-Peer Systems, IPTPS'01, 2002, pp. 251–260. Available at http://dl.acm.org/citation.cfm?id=646334.687813
12 Lesniewski-Laas, C.: 'A sybil-proof one-hop dht'. Proc. of the 1st Workshop on Social Network Systems, SocialNets'08, 2008, pp. 19–24. Available at http://doi.acm.org/10.1145/1435497.1435501
13 Lesniewski-Laas, C., Kaashoek, M.F.: 'Whanau: a sybil-proof distributed hash table'. Proc. of the 7th USENIX Conf. on Networked Systems Design and Implementation, NSDI'10, 2010, pp. 8–8. Available at http://dl.acm.org/citation.cfm?id=1855711.1855719
14 Biryukov, A., Pustogarov, I., Weinmann, P.-P.: 'Detection, measurement and deanonymisation'. Proc. of IEEE Symp. on Security and Privacy, 2013, pp. 80–94
15 Christin, N.: 'Traveling the silk road: a measurement analysis of a large anonymous online marketplace'. Proc. of the 22nd Int. Conf. on World Wide Web, 2013
16 Samarawickrama, S., Jayaratne, L.: 'Automatic text classification and focused crawling'. Sixth Int. Conf. on Digital Information Management (ICDIM), 2011
17 Hermes Center for Transparency and Digital Human Rights: 'Globaleaks platform', 2014. Available at https://globaleaks.org/
18 Freedom of the Press Foundation: 'Securedrop platform', 2014. Available at https://pressfreedomfoundation.org/securedrop
19 Stone-Gross, B., Cova, M., Cavallaro, L., *et al.*: 'Your botnet is my botnet: analysis of a botnet takeover'. Proc. of the 16th ACM Conf. on Computer and Communications Security, CCS '09, 2009, pp. 635–647. Available at http://doi.acm.org/10.1145/1653662.1653738
20 Hopper, N.: 'Challenges in protecting tor hidden services from botnet abuse'. Proc. of Financial Cryptography and Data Security (FC'14), 2014
21 Johnson, A., Wacek, C., Jansen, R., *et al.*: 'Users get routed: traffic correlation on tor by realistic adversaries'. Proc. of the 20th ACM Conf. on Computer and Communications Security, 2013
22 Kwon, A., AlSabah, M., Lazar, D., *et al.*: 'Circuit fingerprinting attacks: passive deanonymization of tor hidden services'. Proc. of the 24th USENIX Security Symp. (Security), 2015
23 Ling, Z., Luo, J., Wu, K., *et al.*: 'Protocol-level hidden server discovery'. Proc. of the 32nd IEEE Int. Conf. on Computer Communications, 2013