2011 Ninth IEEE/IFIP International Conference on Embedded and Ubiquitous Computing

Using traffic analysis to identify The Second Generation Onion Router

John Barker
School of Computer and
Security Science
Edith Cowan University
Mt Lawley, Western Australia
Email: jebarker@our.ecu.edu.au

Peter Hannay
School of Computer and
Security Science
Edith Cowan University
Mt Lawley, Western Australia
Email: p.hannay@ecu.edu.au

Patryk Szewczyk
School of Computer and
Security Science
Edith Cowan University
Mt Lawley, Western Australia
Email: p.szewczyk@ecu.edu.au

***Abstract*—Anonymous networks provide security for users by obfuscating messages with encryption and hiding communications amongst cover traffic provided by other network participants. The traditional goal of academic research into these networks has been attacks that aim to uncover the identity of network users. But the success of an anonymous network relies not only on it's technical capabilities, but on adoption by a large enough user base to provide adequate cover traffic. If anonymous network nodes can be identified, the users can be harassed, discouraging participation. Tor is an example of widely used anonymous network which uses a form of Onion Routing to provide low latency anonymous communications. This paper demonstrates that traffic from a simulated Tor network can be distinguished from regular encrypted traffic, suggesting that real world Tor users may be vulnerable to the same analysis.**

I. Introduction

The first anonymous digital network, commonly known as MixNet was proposed by Chaum in "Untraceable electronic mail, return addresses, and digital pseudonyms" [1]. This paper introduced a concept integral to many future anonymity providing designs, an intermediate system responsible for delivering messages without the identifying details of correspondents. The intermediate system, referred to as a 'mix' also employed public key cryptography to ensure that eavesdroppers could not obtain delivery information.

This seminal paper spurred research into new techniques for providing anonymity and privacy for digital networks. One of these, The Second Generation Onion Router (Tor) is based on technology originally designed by the U.S. Naval Research Lab in 1996 [2] and enjoys some measure of popularity, with an average of two hundred thousand active users as of March 2011 [3].

II. The Second Generation Onion Router (Tor)

Like a mix, messages sent over an onion routing network were encrypted with their routing information and delivered to an intermediate server for forward delivery. Unlike the mix however, messages delivered using the onion routing network were encrypted multiple times, each 'layer' using a different encryption key and routing instructions. The first node in a chain would only be able to encrypt the routing instructions to deliver the message to the next node. Each node in the sequence decrypting a layer until the complete message is decrypted and transmitted to the destination. Figure 1 shows the path a typical message takes through the Tor network.

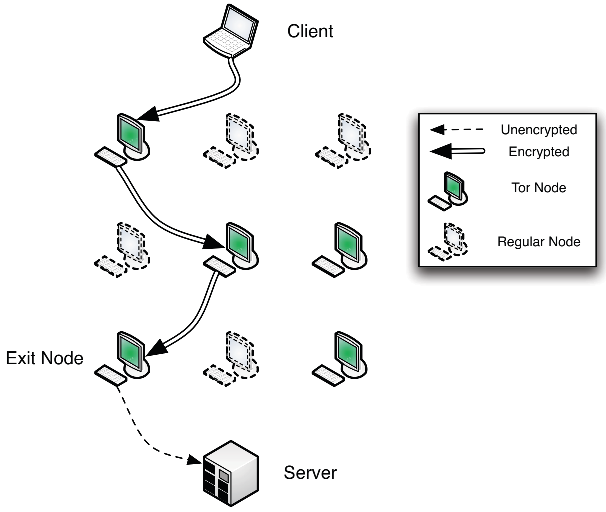

Figure 1. The Tor Network

Traffic enters the Tor network through an onion proxy which accepts TCP streams. Some identifying features are scrubbed from the data using application filters before the data is relayed over the network through TLS [4] encrypted connections. The intermediate nodes responsible for routing messages are known as relays and are typically chained together to construct a circuit. When traffic leaves the Tor network it is delivered by a special kind of relay known as an exit node. At an exit node the data is transmitted in the original format it was supplied to at the onion proxy.

The onion proxy builds circuits incrementally, first obtaining a session key from each successive relay in a circuit. Once all session keys for a circuit have been obtained, the message is broken into fixed sized cells of 512 bytes and iteratively encrypted using the session key of each node in the circuit in the reverse order that the data traverses the network. Cells

978-0-7695-4552-3/11 $26.00 © 2011 IEEE
DOI 10.1109/EUC.2011.76

come in two forms: control cells and relay cells. Control cells are used to create and maintain circuits, while relay cells contain commands for circuit maintenance and additional data for verifying message integrity and identifying streams.

III. Weaknesses and Attacks

When designing a system such as Tor, a number of tradeoffs have to be made between the strength of the security provided and the convenience and performance of the system. Tor designers consciously prioritized low latency, usability and flexibility against security goals such as security against end to end attacks [5, p. 4]. This means that by design, Tor is vulnerable to a global passive adversary, however there are some attacks that were not expected by the designers.

A well known attack involves sniffing traffic that leaves exit nodes to capture private information [6]. Many users assume that Tor provides end to end encryption, and transmit private information over the Tor network.

Technologies such as Javascript and Flash can be embedded in web pages accessed by Tor users, and have control over network resources. By injecting network traffic with certain patterns alongside regular network traffic, Tor users can be identified [7].

Tor bridges, intended as a way to get censored users access to Tor are easily identified. They are also vulnerable to clogging attacks which make bridge operators more easily identified [8].

Murdoch and Danezis [9] proposes a technique to identify users by estimating the latency of individual Tor nodes using a hostile Tor node. The hostile Tor node is able to send data to users using a predictable traffic pattern and identify this pattern as it is repeated through the network, correlating streams back to individuals.

This attack was shown to be impractical as the Tor network grew in size, with the increased number of users adding enough congestion to mask the introduced identifying patterns [10]. However a weakness in the Tor design meant that particularly configured hostile nodes could amplify the deliberately introduced congestion to make individuals identifiable on the larger network.

IV. Significance

Most conventional attacks against secure networks are known as traffic analysis, this is the process of examining information about the communications rather than the information contained within them. This information may include the size of messages communicated, their frequency and timing. Many researchers have proposed traffic analysis techniques that allow attackers to reveal the identities of Tor users.

While Tor has been the subject of much academic literature [11–13], the primary objective of researchers has historically been the attempt to reveal participant identities [9, p. 3]. Many attack techniques proposed have been somewhat academic in nature and not necessarily feasible in practice [14]. In certain circumstances they require compromise of large parts of the Tor network, supplying hostile data to Tor users or complicated knowledge of usage patterns and an excess of patience. The technique demonstrated in this paper is a low cost technique, which does not require sophisticated equipment and can be completed by a passive observer.

V. Traffic Classification

When considering the use of traffic analysis for classification of Internet communications, three techniques are used: exact matching, heuristic matching and machine learning [15]. Since Tor employs strong encryption and can communicate on any port, it can easily an exact matching technique through simple configuration options.

Heuristic based techniques have been designed to classify encrypted communications, including the identification of P2P traffic [16–18] and viruses [19].

Machine learning algorithms have also been used successfully to classify encrypted traffic including Skype [20] and to identify application protocols tunnelled over SSH [21]. An previous attempt to classify Tor using Bayesian networks was attempted in Herrmann et al. [13] without great success.

It is difficult to say what machine learning technique is the most effective as no consensus has been reached, the literature covers a wide variety of techniques each with vastly different goals and no two techniques can be directly compared as the data used for analysis has not been disclosed [22]. However some attempt has been made at comparison in Williams et al. [23] which suggests that the C4.5 algorithm has the greatest performance and accuracy when compared to a number of Bayesian algorithms. Mohd [24] compares thirty machine learning algorithms to find random tree, IB1, IBK and random forest algorithms obtaining the greatest classification accuracy.

VI. Experiment

To determine if Tor traffic can be distinguished from regular encrypted traffic, an experiment was conducted to generate a series of traces for comparison. Network traffic was generated using the commonly available Firefox browser, version 3.6.8 installed on an Ubuntu 10.04 desktop operating system. One hundred and seventy simulations were run of varied user interactions against a sample of thirty websites, using version 1.0.6 of the Selenium Browser testing framework. A private Tor network was configured running three directory servers and fifteen relays, with version 0.2.1.26.

The experiment consisted of three phases, the first was the capture of regular HTTPS traffic which began Sunday the 3rd of October 2010 and finished Sunday the 17th that same month. Phase two began immediately after and continued two weeks till the 31st of October. This phase involved capturing regular HTTP traffic routed through a private Tor network. The final phase began on the 7th of November and concluded on the 21st, this phase was the capture of HTTPS traffic through a private Tor network.

To reduce the affect of confounding variables, all phases conducted the same simulations, on an isolated test network and were conducted within a virtual machine snapshot which was periodically rolled back to a clean, known state. The experiment yielded three sets of, a summary is included in table I.

Phase	Total Size	Packets	Sessions
HTTPS	9.52GB	11,883,703	236,659
HTTP over Tor	10.50GB	14,823,849	168,876
HTTPS over Tor	5.58GB	8,161,620	95,203

Table I
CAPTURED DATA

VII. RESULTS

The data captured was in the form of 1MB capture files which were recombined with mergecap [25] and processed by NetAI [26] to produce ARFF format files for use by Weka [27, 28]. NetAI identifies flows inside the capture files and produces a number of statistics to be used as attributes for classification.

All of the algorithms chosen were able to successfully classify HTTPS and HTTP over Tor traffic with accuracy in excess of 90%, with the Adaboost algorithm unable to classify HTTPS over Tor. The best performing algorithm random forest was able to classify HTTP over Tor with 93.7% accuracy and a false positive rate of 3.7%. HTTPS over Tor was more easily identified with a 97.7% accuracy and low false positive rate of 0.3%. A summary of the results obtained by these machine learning classifiers is available in Table II.

	True Positive Rate	False Positive Rate	ROC Area
Random Forest			
HTTPS	0.957	0.036	0.99
HTTP over Tor	0.937	0.037	0.986
HTTPS over Tor	0.977	0.003	0.999
Weighted Avg.	0.954	0.03	0.99
j4.8 With 10 fold cross validation			
HTTPS	0.951	0.04	0.989
HTTP over Tor	0.978	0.043	0.98
HTTPS over Tor	0.97	0.007	0.992
Weighted Avg.	0.964	0.018	0.986
Adaboost			
HTTPS	0.95	0.001	0.975
HTTP over Tor	0.999	0.324	0.838
HTTPS over Tor	0	0	0.777
Weighted Avg.	0.785	0.109	0.891

Table II
RESULTS FROM MACHINE LEARNING CLASSIFIERS

With some relative success classifying Tor traffic using unsupervised machine learning techniques, the captures were examined for a heuristics based classification approach. When a histogram of packet sizes for each of the packet traces is produced, a clear distinction can between seen between the sample sets as seen in Figures 2, 3 and 4.

Investigating more closely, by examining individual packet traces some significant patterns begin to appear. Table III shows the packet sizes seen for the first 6 packets of a 1% sample of sessions in each phase of data captured. The first few packets contain 0 bytes of data, these are the typical SYN, SYN/ACK and ACK flagged packets that are used to initiate a TCP/IP connection. Following that only two packet sizes are seen for HTTPS connections. For Tor sessions a range of packet sizes from 131 to 152.

Index in Stream	Observed Packet Sizes
HTTPS	
0	0
1	0
2	0
3	100, 110
4	0
5	122, 516
HTTP Over Tor	
0	0
1	0
2	0
3	131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152
4	0
5	915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934
HTTPS Over Tor	
0	0
1	0
2	0
3	131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 144, 145, 146, 147, 148, 149, 150, 151, 152
4	0
5	914, 915, 916, 918, 919, 920, 921, 922, 923, 925, 926, 927, 928, 929, 930, 932, 934, 935, 936

Table III
PACKET SIZES

This knowledge can be used to build a rudimentary classification algorithm to identify Tor traffic, the pseudo code for an example algorithm is included in Figure 5.

```
if  packet[3] > 130 and
    packet[3] < 153 and
    packet[5] > 913 and
    packet[5] < 937 then
  is_tor = true
else
  is_tor = false
end
```

Figure 5. Pseudo Code for Matching a Tor Session

Applying this algorithm to the complete traces yields the results as seen in Table IV.

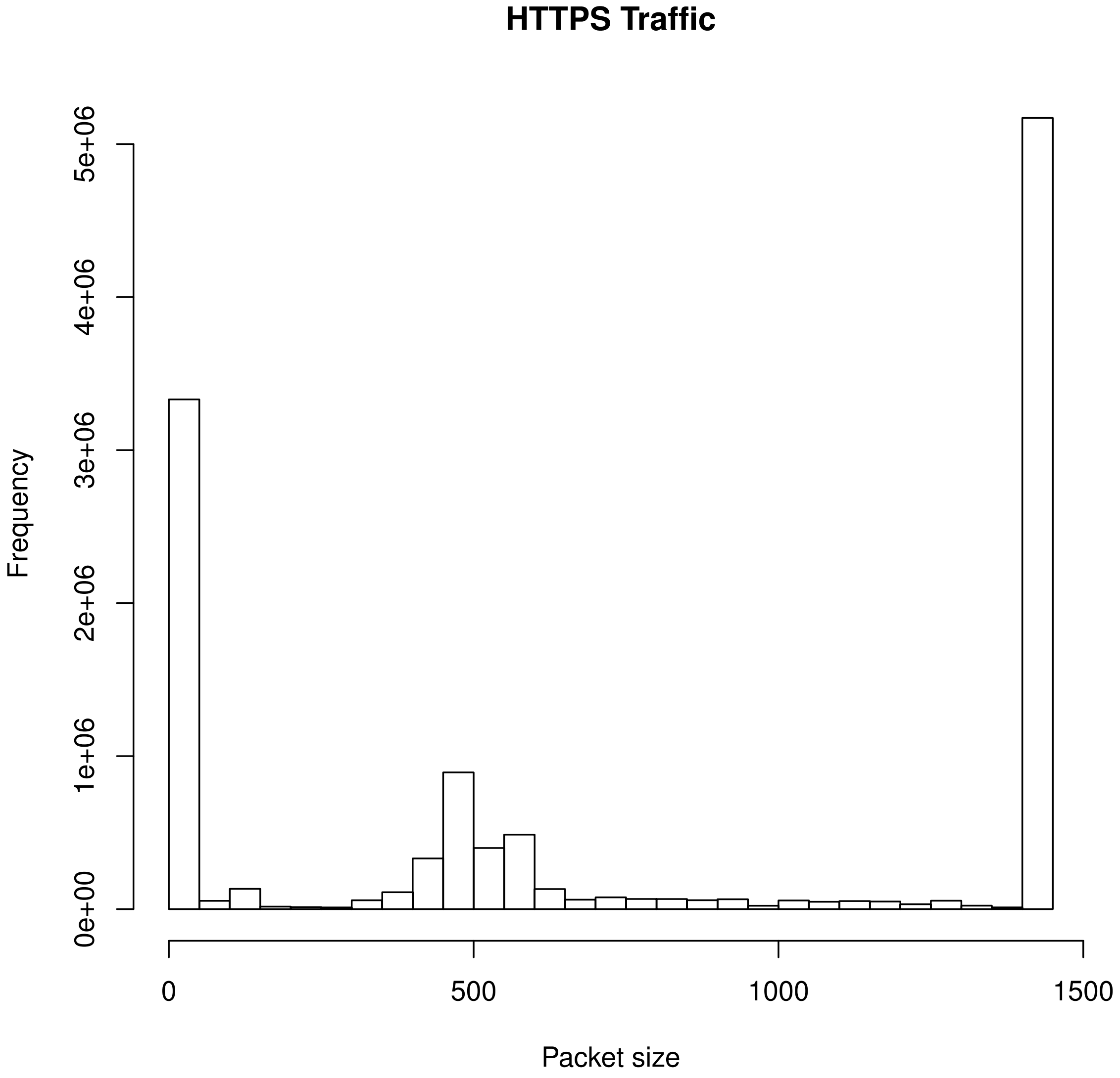

Figure 2. Histogram of packet size for HTTPS traffic

Protocol	Identified as Tor
HTTPS	1.06%
HTTP over Tor	98.13%
HTTPS over Tor	97.54%

Table IV
RESULTS FROM HEURISTIC CLASSIFIER

When examining the cases where the algorithm failed to identify Tor streams, in all cases the session had failed to successfully initiate the handshake required for a TCP session. Which meant that no meaningful data could have been transmitted.

VIII. DISCUSSION

Using Tor as a communications proxy incurs some overhead, sufficient enough that when using a Tor proxy in controlled conditions, its traffic can be distinguished from regularly encrypted traffic. This overhead may be sufficient enough that Tor nodes in a real world network can be identified by using readily available eavesdropping techniques.

The encryption layers that wrap Tor level communications, do not appear to obfuscate the size of communications sent between Tor nodes. This is most apparent in the composition of packet sizes that make up an individual session, with Tor sessions showing a large percentage of packets just large enough to fit the 512 byte cells that make up the Tor protocol.

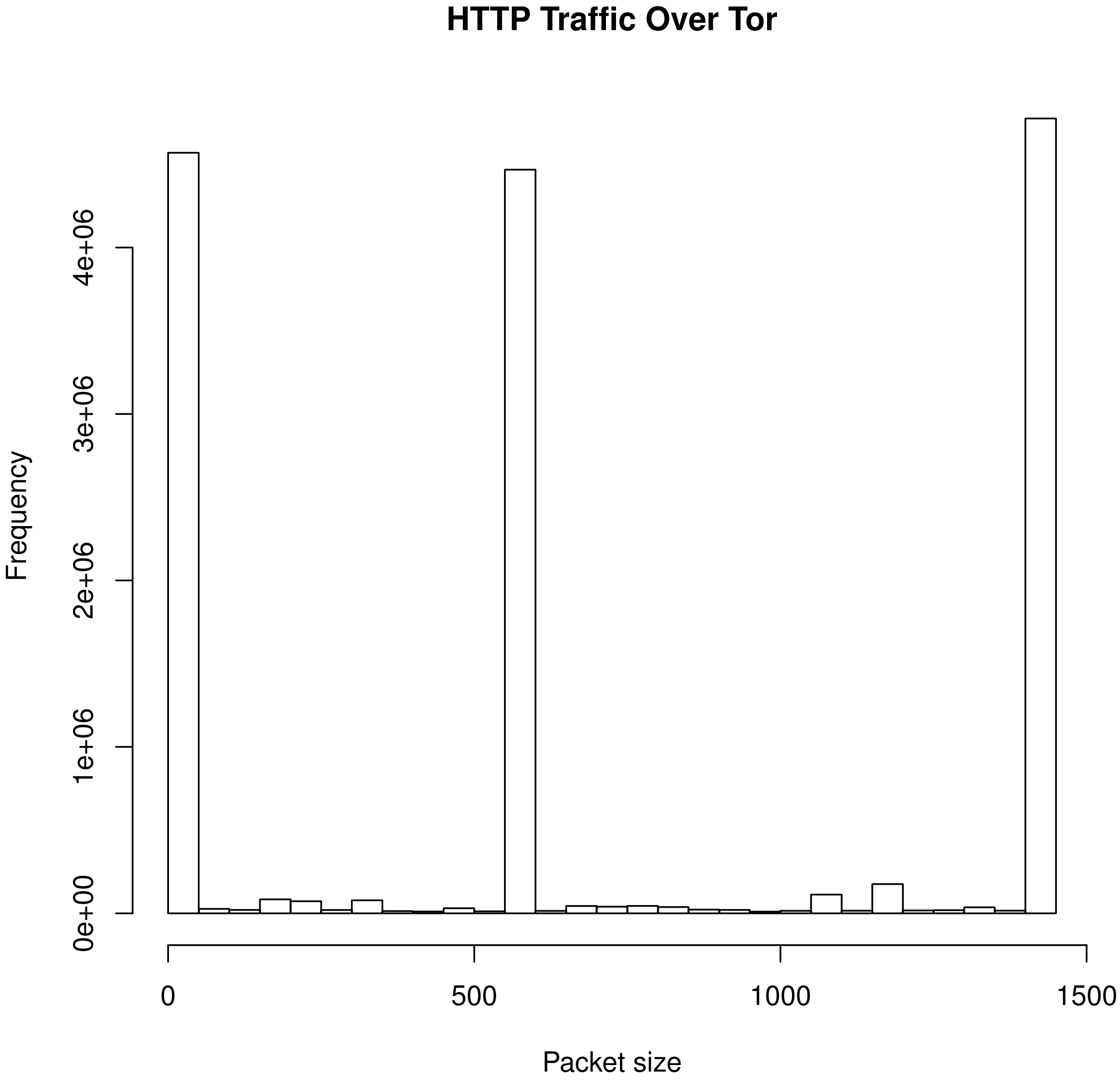

Figure 3. Histogram of packet size for HTTP traffic over Tor

However, this experiment was based on a small set of simulated data, with which it would be impossible to cover all possible real world conditions. The variability and noise present in the real Tor network may make this classification technique impossible.

IX. Conclusion

This research demonstrates that Tor does have characteristics that make it distinguishable from regular encrypted traffic. The encryption used by Tor does not appear to blur the size of communication cells sufficiently to prevent automated identification of Tor traffic, even with only a few observed packets. While the scope of this research is limited, it suggests that it may be possible to build simple software to automatically identify Tor users and block them from the network.

Further research needs to be conducted with live packet traces from real participants in the Tor network. Most existing traffic classification research operates this way, using exact matching techniques to separate traces collected into treatment and control groups. Real Tor traffic might be captured from publicly available and co-operating Tor nodes and compared to real encrypted sessions between well known public HTTPS servers.

Training a classifier against real world traffic will account for the different and varied nature of packet switching hardware, with different maximum transmission units, performance capabilities and geographic location.

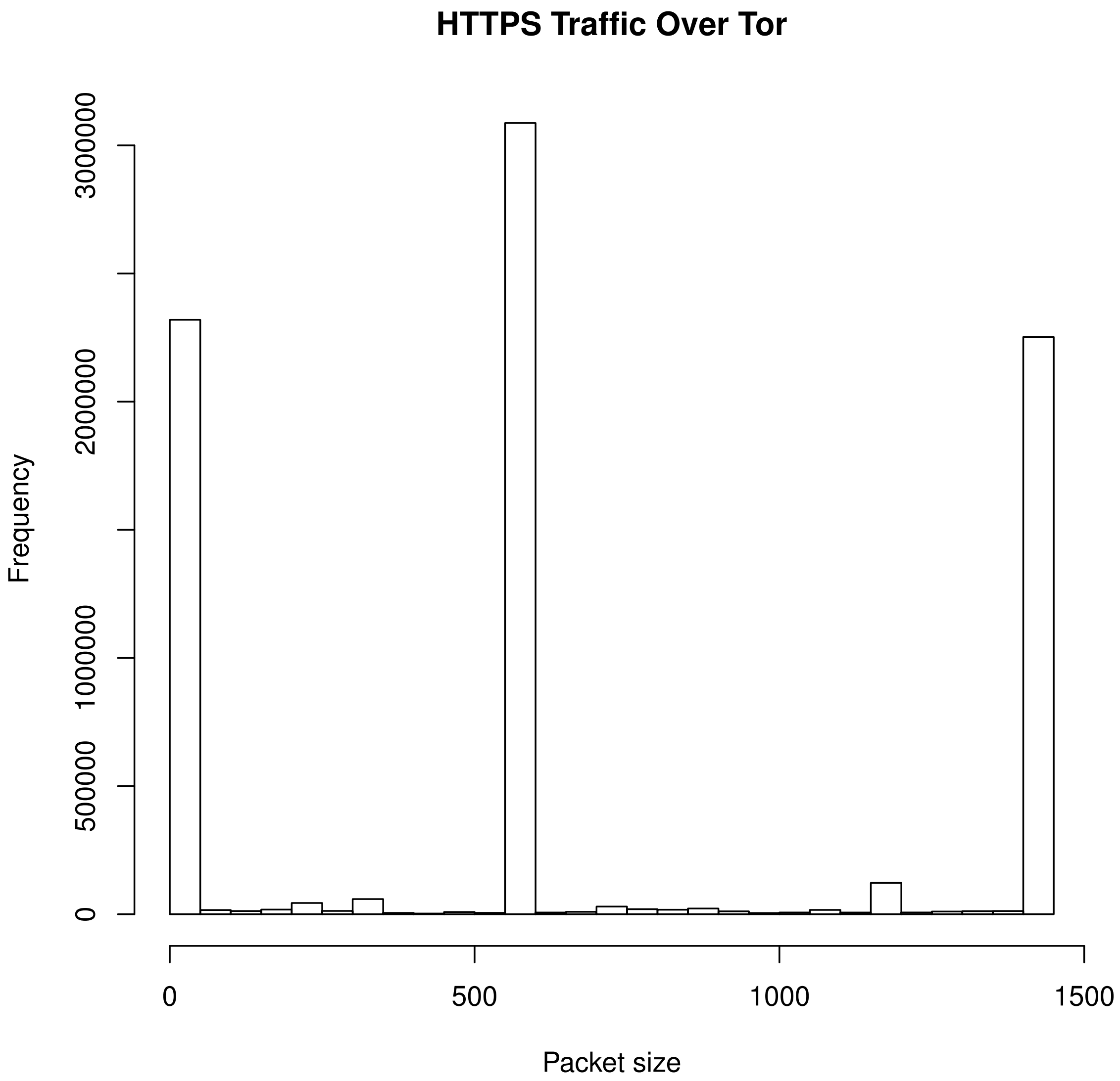

Figure 4. Histogram of packet size for HTTPS traffic over Tor

The heuristic based classifier used in this paper used a single attribute: the size of individual packets in a stream, to classify with great accuracy. But other attributes may be considered, the most typical of these being inter packet arrival time - though this is likely to be more affected by natural variability.

The encrypted traffic generated in the first phase of the experiment, only covered a single application and version. It may be possible that other applications use the same protocol with encryption and have characteristics similar to that produced by Tor. This would lead to any proposed classifier generating false positives. Further research should also consider the nature and characteristics of a wider set of encrypted communications.

References

[1] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Communications of the ACM*, vol. 24, no. 2, pp. 84 –88, 1981.

[2] D Goldschlag and M Reed, “Hiding routing information,” *Information Hiding*, 1996.

[3] The Tor Project, Inc., *Tor metrics portal: users*, Retrieved June 2011 from https://metrics.torproject.org/users.html, June 2011.

[4] E. Rescorla and T. Dierks, *The transport layer security (TLS) protocol*, Retrieved March, 2010 from http://tools.ietf.org/html/rfc5246, 2008.

[5] R Dingledine, N Mathewson, and P Syverson, "Tor: the second-generation onion router," *Proceedings of the 13th conference on USENIX Security Symposium-Volume 13*, p. 21, 2004.

[6] D. Goodin, *Tor at heart of embassy passwords leak*, Retrieved March, 2010 from http://www.theregister.co.uk/2007/09/10/misuse_of_tor_led_to_embassy_password_breach/, 2007.

[7] T Abbott, K Lai, M Lieberman, and E Price, "Browser-based attacks on Tor," *Privacy Enhancing Technologies*, pp. 184–199, 2007.

[8] J McLachlan and N Hopper, "On the risks of serving whenever you surf: vulnerabilities in tor's blocking resistance design," *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pp. 31–40, 2009.

[9] S. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," *IEEE Symposium on Security and Privacy. IEEE CS*, 2005.

[10] N. Evans, R Dingledine, and C Grothoff, "A practical congestion attack on tor using long paths," *18th USENIX Security Symposium*, pp. 33–50, 2009.

[11] N Hopper, E. Vasserman, and E Chan-Tin, "How much anonymity does network latency leak?" *Proceedings of the 14th ACM conference on Computer and communications security*, p. 91, 2007.

[12] S. Murdoch and P Zielinski, "Sampled traffic analysis by internet-exchange-level adversaries," *Lecture Notes in Computer Science*, vol. 4776, p. 167, 2007.

[13] D Herrmann, R Wendolsky, and H Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier," *Proceedings of the 2009 ACM workshop on Cloud computing security*, pp. 31–42, 2009.

[14] The23rd Raccoon, *How I learned to stop ph34ring NSA and love the base rate fallacy*, [Electronic mailing list message]. Retrieved March 2011 from http://archives.seul.org/or/dev/Sep-2008/msg00016.html, January 2008.

[15] M Zhang, W John, K Claffy, and N Brownlee, "State of the art in traffic classification: a research review," *PAM Student Workshop*, 2009.

[16] T Karagiannis, A Broido, and M Faloutsos, "Transport layer identification of p2p traffic," *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 121–134, 2004.

[17] M Perényi, T. Dang, A Gefferth, and S Molnár, "Identification and analysis of peer-to-peer traffic," *Journal of Communications*, vol. 1, no. 7, p. 36, 2006.

[18] W John and S Tafvelin, "Heuristics to classify internet backbone traffic based on connection patterns," *Information Networking*, January 2008.

[19] A Lazarevic, L Ertoz, V Kumar, A Ozgur, and J Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," *Proceedings of the Third SIAM International Conference on Data Mining*, pp. 25–36, 2003.

[20] R Alshammari and A. Zincir-Heywood, "Machine learning based encrypted traffic classification: identifying ssh and skype," *Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications*, pp. 289–296, 2009.

[21] M Dusi, F Gringoli, and L Salgarelli, "A preliminary look at the privacy of ssh tunnels," *Computer Communications and Networks*, pp. 1–7, 2008.

[22] H Kim, U. CAIDA, D Barman, and M Faloutsos, "Comparison of internet traffic classification tools," *ANF Workshop*, vol. 2, 2007.

[23] N Williams, S Zander, and G Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 5, p. 16, 2006.

[24] A. Mohd, "Towards a flow-based internet traffic classification for bandwidth optimization," *International Journal of Computer Science and Security (IJCSS)*, vol. 3, no. 2, p. 146, 2009.

[25] S. Renfro, *Mergecap - the wireshark network analyzer 1.5.0*, Retrieved August 2011 from http://www.wireshark.org/docs/man-pages/mergecap.html.

[26] Swinburne University of Technology, *netAI network traffic based application ddentification*, Retrieved May 2011 from http://caia.swin.edu.au/urp/dstc/netai/, August 2006.

[27] M Hall, E Frank, G Holmes, B Pfahringer, P Reutemann, and I. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[28] R. Bouckaert, E Frank, M Hall, and R Kirkby, "WEKA Manual for Version 3-7-2," 2010.