Torben: A Practical Side-Channel Attack for Deanonymizing Tor Communication

Daniel Arp
University of Göttingen
Göttingen, Germany

Fabian Yamaguchi
University of Göttingen
Göttingen, Germany

Konrad Rieck
University of Göttingen
Göttingen, Germany

Abstract

The Tor network has established itself as de-facto standard for anonymous communication on the Internet, providing an increased level of privacy to over a million users worldwide. As a result, interest in the security of Tor is steadily growing, attracting researchers from academia as well as industry and even nation-state actors. While various attacks based on traffic analysis have been proposed, low accuracy and high false-positive rates in real-world settings still prohibit their application on a large scale.

In this paper, we present *Torben*, a novel deanonymization attack against Tor. Our approach is considerably more reliable than existing traffic analysis attacks, simultaneously far less intrusive than browser exploits. The attack is based on an unfortunate interplay of technologies: (a) web pages can be easily manipulated to load content from untrusted origins and (b) despite encryption, low-latency anonymization networks cannot effectively hide the size of request-response pairs. We demonstrate that an attacker can abuse this interplay to design a side channel in the communication of Tor, allowing short *web page markers* to be transmitted to expose the web page a user visits over Tor. In an empirical evaluation with 60,000 web pages, our attack enables detecting these markers with an accuracy of over 91% and no false positives.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and Protection*; I.5.4 [**Pattern Recognition**]: Applications

Keywords

Anonymity; Side Channels; Traffic Analysis

ASIA CCS'15, April 14–17, 2015, Singapore..
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3245-3/15/04 ...$15.00.
http://dx.doi.org/10.1145/2714576.2714627.

1. INTRODUCTION

The Tor network is one of the largest efforts to provide anonymity and privacy on the Internet. The network implements a low-latency anonymity service based on the concept of Onion Routing. The network consists of over 6,000 relay nodes worldwide that enable its users to relay communication through a circuit of these nodes, for example, to anonymously express their opinion or circumvent digital censorship.

With the increasing use of Tor in practice, research on attacks against this service have gained considerable attention. A large body of work has studied *passive attacks* based on traffic analysis, most notably website fingerprinting and traffic confirmation attacks. The former enable an attacker to detect patterns indicative for web pages in Tor traffic [e.g., 2, 4, 8]. Although these approaches provide good results in closed-world settings, in practice they suffer from high false-positive rates. Traffic confirmation attacks on the other hand are not limited to web traffic but require an attacker who is able to eavesdrop on both ends of a communication over a long period of time [e.g., 6, 9]. A second strain of research has thus considered *active attacks* against the Tor network, such as path-selection attacks based on network congestion and traffic watermarking using packet delays [e.g., 10]. While these techniques provide a more accurate and faster deanonymization, they involve a significant effort for the adversary. As a result, the few known cases of deanonymization of Tor have been reported to instead make use of advertisement networks or rely on vulnerabilities in browser implementations and are thus unrelated to insecurities of Tor itself.

In this paper, we present *Torben*, a novel deanonymization attack against Tor that is significantly more reliable than traffic analysis attacks but far less intrusive than browser exploits. In contrast to other active attacks, Torben operates entirely on the application layer and does not require Tor nodes or routers to be controlled by the adversary. The attack is based on an unfortunate interplay of technologies: First, web pages can often be manipulated to load content from untrusted origins, for example, using advertisements or user-provided content. Second, despite encryption, low-latency anonymization networks cannot effectively hide the size of request-response pairs in web traffic. We show that an attacker can abuse this interplay to design a side channel in the communication of Tor. This side channel enables the transmission of short *web page markers* that expose the web page a user visits to an observer between the Tor client and the entry node. Although it is well-known that active web content allows to track the visitors of web pages, we are the first to show that it can be used to deanonymize Tor users in a short period of time.

In summary, our contributions are the following:

- We present a novel side channel in Tor communication. By issuing HTTP requests from the user's browser, an attacker is able to induce distinct patterns observable in encrypted traffic.
- We demonstrate that this side channel can be used to perform a novel deanonymization attack against Tor, allowing us to transmit *web page markers* exposing the page visited by a user.
- Finally, we show that these web page markers can be accurately detected in real Tor traffic with high accuracy by combining techniques from signal processing and machine learning.

2. BACKGROUND

Before presenting our deanonymization attack and discussing details of how to transmit data through the underlying side channel, we need to briefly review the basics of the Tor network (Section 2.1) and define the attack scenario we are considering (Section 2.2).

2.1 The Tor Network

The Tor network [3] is a low-latency anonymization network whose purpose is to protect the privacy of its users by obfuscating their network traffic. This is achieved by tunneling user traffic through arbitrary paths in the Tor network, which consist of multiple hops (Tor relays) that run the Tor software and are operated by volunteers. The security of Tor is based on the use of strong encryption and the large number of relays that can be used to establish a path, thus significantly lowering the ability of an attacker to easily eavesdrop a communication or link senders and receivers.

A user who wants to establish a connection to a server through Tor has to run a Tor client on his computer which will first select a path through the Tor network. After establishing a path, the user can send data over the Tor network using fixed size Tor cells which are multi-layer encrypted with previously negotiated session keys. Each relay node on the path then removes one layer of encryption while the cell is forwarded to its destination.

2.2 Attack Scenario

For our attack, we consider a scenario that involves an active attacker. We assume that, first, this attacker is able to monitor the encrypted communication between a Tor client and the entry node, and, second, she actively implants a marker into a web page of interest. An overview of this attack scenario is depicted in Figure 1.

Let us, for instance, consider a totalitarian regime or law enforcement agency that wants to determine whether a particular user visits a certain web page, despite the fact that Tor is being used to anonymize the communication. Clearly, this attacker can be expected to be capable of observing the encrypted network communication between the user's browser and the Tor entry node. However, this alone is a vast underestimation of her capabilities, as it considers a passive attacker. It is reasonable to assume that the user's browser may be exposed to attacker-provided web content at some point throughout the browsing session. This content may be delivered through a multitude of vectors. Based on the chosen vector, we consider the following two variants of the attack scenario:

- **Remote markers** In this scenario, the attacker exploits the fact that web pages often embed content from different origins, some of which might be controlled by the attacker. The attacker may, for instance, be able to host and advertisement on the web page.
- **Local markers** In this scenario, the attacker is able to inject content directly into a web page. For example, the content of a web page may be manipulated at the server to track its users.

Regardless of the type of the marker, attacker-provided content is loaded in the user's browser and can be used to generate a characteristic pattern in the resulting Tor communication. This pattern can then be detected in the encrypted traffic between the Tor client and the entry node, ultimately enabling an adversary to deanonymize the visitors of marked web pages.

3. A SIDE-CHANNEL ATTACK ON TOR

A fundamental limitation of low-latency anonymization networks is that they cannot effectively hide the sizes and order of relayed packets. For users browsing web pages via Tor, this means that HTTP request and response sizes directly influence the stream of TLS records observed between the Tor client and the entry node. Unfortunately, this setting can be exploited by an attacker. If the user accesses attacker-controlled content, such as JavaScript code, the attacker gains partial control over the stream of request and response sizes.

The overall idea of our attack is to leverage this control to carry out a side-channel attack by creating distinct communication patterns in the encrypted data stream that can be effectively detected using machine learning techniques. While this idea is simple at core, applying it to construct a successful attack requires careful engineering of a number of different components. In particular, the following four challenges need to be addressed:

- **Preprocessing of network traces.** Network traces need to be preprocessed and transformed into a robust representation suitable for analysis of request and response sizes (Section 3.1).
- **Side channel design.** A reliable side channel needs to be designed that allows short messages to be transmitted to an attacker observing the encrypted data stream of Tor (Section 3.2).
- **Transmission of web page markers.** Markers need to be transmitted using the side channel, such that the visit of a marked web page induces a distinct pattern in the encrypted traffic (Section 3.3).
- **Detection of web page markers.** Finally, a method for automatic detection of these web page markers is required that enables identifying individual markers in real network traces (Section 3.4).

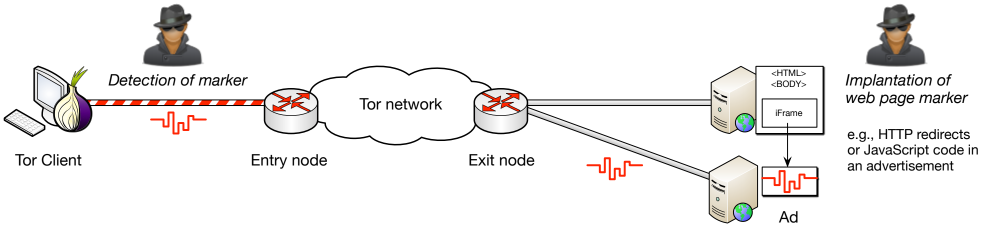

Detection of marker
Tor network
Tor Client
Entry node
Exit node
<HTML>
<BODY>
iFrame
Implantation of web page marker
e.g., HTTP redirects or JavaScript code in an advertisement
Ad

Figure 1: Attack scenario: A web page marker is implanted using embedded or user-provided content, such as an advertisement. The marker induces a traffic pattern visible at the entry node, for example, using JavaScript code.

3.1 Preprocessing of Network Traces

The success of traffic analysis attacks critically depends on the choice of a suitable representation of observed network communication. With this goal in mind, we preprocess network traces by leveraging inherent properties of the Tor protocol and the protocols it depends on, thus allowing us to remove noise sources and highlight those properties of network traffic that are controllable by the attacker. The key insight our preprocessing scheme is based on, is that by controlling the size of HTTP requests and responses, we do not gain control over the size of IP packets or TLS records but only over the *amount* of data transferred from one change of direction to the next. We devise a representation emphasizing this aspect of the record stream in a two-step procedure outlined in the following.

3.1.1 TCP Stream Reassembly

All Tor communication takes place via Tor cells encapsulated in TLS records. While HTTP communication relayed by the Tor network may influence these sequences, they are distorted by several noise sources that needlessly complicate analysis. In particular, delayed or dropped IP packets cause the transport layer to issue re-transmissions of packets.

Fortunately, we can easily address all of these problems by reassembling TCP streams using readily available tools such as *tshark*. This allows all subsequent analysis to be carried out on streams of TLS records as opposed to raw IP packets. In effect, the order of Tor cells is preserved and artifacts of TCP/IP such as re-transmissions and acknowledgments carrying no data are removed. To simplify all further processing, we map traces to sequences of record sizes where a positive and negative sign are used for incoming and outgoing traffic respectively.

3.1.2 Filtering and Merging TLS Records

We proceed to apply the following chain of transformations to account for various properties of Tor and TLS that impact the analysis of network traffic.

Filtering empty records. As a first step, we filter sequences such that they only contain entries representing records of 100 bytes or more. This preserves Tor cells as the minimum cell size is 512 bytes.

Merging of records. We merge adjacent TLS records going into the same direction to obtain a sequence representing the amount of data rather than individual records.

Filtering control cells. We filter single cells that are unrelated to user data by discarding any sizes smaller than twice the cell size after merging. Upon removal of these single cells, we merge TLS records again to connect data relay cells previously separated by control cells.

Normalization of sizes. The concrete sizes of transfered data vary slightly depending on the version of Tor and its libraries. As a simple normalization, we express all sizes in multiples of 2000 bytes.

Upon completion of the preprocessing step, each Tor connection is represented by a sequence of integers that encodes the amount, direction and order of data transfers. We refer to these sequences as *data transfer sequences* throughout the rest of this paper.

3.2 Side Channel Design

We design our side channel such that it satisfies the following two properties: First, normal web traffic should be clearly distinguishable from any side-channel communication to make false positives very unlikely. Second, the transmission speed needs to be high enough to allow short byte sequences to be transmitted before the user leaves a web page.

Side-channel communication can only be distinguished from regular web traffic, if data transfer patterns exist that are atypical for normal web pages. We thus design 16 different symbols using request-response pairs with discrete request and response sizes of $2,000$, $4,000$, $6,000$ and $8,000$ bytes. These sizes are large enough to be atypical but still enable a fast transmission.

A message is then encoded by splitting it into quad bits and encoding each quad bit separately using the corresponding request-response pair. By concatenating the resulting symbols, we finally obtain a data transfer sequence suitable for transmission over the side channel. A more detailed discussion of our side channel can be found in [1].

3.3 Transmission of Web Page Markers

Equipped with a side channel, we can now expose visited web pages by transmitting suitable messages from the user's browser. This, however, creates two additional challenges. First, a suitable browser-based mechanism for transmission of correctly ordered sequences of HTTP requests needs to be found, and second, *web page markers* that encode the names or URLs of visited web pages need to be constructed.

3.3.1 Issuing HTTP Requests

In our experiments, we assume that an attacker is able to execute JavaScript code within the browser of a Tor user. This code can be embedded in a displayed advertisement, injected via cross-site scripting or contained in any other included JavaScript code. For establishing the side channel, the code does not need to operate in the context of the marked web page and thus our attack is not effected by the same-origin policy.

The standard JavaScript object `XMLHttpRequest` offers a mechanism for request transmission that fits our needs perfectly. The object allows requests to be issued from the user's browser synchronously while offering fine-grained control over request content and headers. We can employ `XMLHttpRequest` to transmit a request-response pair (r_1, r_2) using the following URL

```
http://server.com/res?str
```

where `res` is simply a resource of size r_1, such as an image or a document, and `str` a random string of length r_2 attached to the URL. We choose a random string here to reduce caching effects induced by some browsers. Note that the contacted server can be any server offering resources with the required sizes. It is sufficient to find a server that offers resources with four different sizes to successfully carry out the attack. Consequently, the server does not need to be controlled by the attacker and hence provides no information about the attacker's origin.

3.3.2 Web Page Markers

We are now able to transmit short messages over the side channel to expose web pages visited using Tor. If the attacker only wants to monitor a single web page, choosing a constant message is sufficient. However, to monitor multiple web pages or track how a user browses from one page to another, the web page markers need to be designed such that they have a large pairwise Hamming distance. To achieve this, we use the 20 byte SHA-1 hash value of the monitored URL as a web page marker. This ensures that there is a natural mapping between URLs and their markers.

3.4 Detection of Web Page Markers

The presented side channel allows web page markers to be transmitted and recognized by human observers, yet due to the vast amount of network traffic to analyze, our attack only becomes practical if the markers can be detected automatically.

To identify markers in a monitored data transfer sequence, we use a sliding window and classify the sequence under each window independently. This highlights an advantage of our attack over related traffic analysis attacks. Since the marker is transmitted in a small time frame, it is perfectly valid to limit the analysis to small sequences of the network trace, in our experiments 100 symbols, thus significantly reducing the computational resources required for training a classifier and detecting web page markers.

In order to detect the markers, we train a multi-class Support Vector Machine (SVM) with probabilistic outputs on the sequences of individual web page markers. Given an unknown sequence, the SVM can then be used to identify the most likely web page marker present in this sequence. Due to the probabilistic output, we are able to quantify the confidence of this decision, which enables us to determine sequences that do not correspond to any of the known markers. A sequence is thereby mapped into a high-dimensional vector space using positional n-grams which we introduced for this purpose. More details on the embedding can be found in the technical report [1].

4. EVALUATION

We evaluate the Torben attack in a series of experiments that allows us to assess its effectiveness in different settings. After presenting our experimental setup (Section 4.1 & 4.2), we first evaluate the attack in a closed-world setting (Section 4.3) where users can only visit web pages from a fixed set. We furthermore proceed to consider an open-world setting where users can freely choose web pages from a potentially infinite number of web pages—60,000 in our experiment—(Section 4.4). The open-world setting allows us to obtain a good approximation for the false-positive rate of our attack. Finally, we perform a live experiment where we evaluate the ability of Torben to identify web page markers in real-world traffic generated by multiple users (Section 4.5). Note, that a more detailed description of our evaluation can be found in [1].

4.1 Data Collection

To automatically visit a large number of web pages over the Tor network in an acceptable time, we use the *Selenium WebDriver* (version 2.38.3), a browsing automation plug-in for Mozilla Firefox. The plug-in is installed along side the Tor browser bundle (version 3.5), the standard distribution of Tor. This ensures that the browser configuration used in our experiments almost exactly matches the configuration employed by most Tor users.

For our experiments we automatically visit different sets of the top one million web pages from the Alexa ranking (retrieved in February 2014). For the vast majority of web pages, the *Selenium WebDriver* can automatically determine when the page is fully loaded based on loading of the page icon. In rare cases, successful page load is not detected within 3 minutes. In these cases we discard this web page and use the next page in the Alexa ranking instead. Moreover, we remove variants of very similar pages. For example, we consider only `google.com` and not `google.de` or `google.fr`. This is necessary to obtain a better comparison with website fingerprinting approaches as these approaches are known to fail if web pages are too similar.

4.2 Detection Setup

To perform the Torben attack in practice, we implement the preprocessing, transmission and detection steps outlined in Section 3. For extracting positional n-grams from network traces, we use the tool *Sally*[1] (version 0.8.3) and for learning the probabilistic classification the library *LibSVM*[2].

For training the detection method described in Section 3.4, we generate 100 web page markers using the SHA-1 hashes of the top Alexa web pages and record 50 transmissions for each of these markers over Tor. Note that only the web page markers and not the actual web pages are recorded for training our detection method.

4.3 Closed-World Evaluation

In the first experiment, we consider a closed-world setting, where we visit each of the top 100 Alexa web pages 50 times, first in February 2014 and a second time in April 2014 resulting in two data sets. To simulate the attack scenario of a remote marker, we use a reverse proxy and inject a small JavaScript snippet into each page that opens a sep-

[1] http://www.mlsec.org/sally/

[2] http://www.csie.ntu.edu.tw/c̃jlin/libsvm/

arate browser window containing the marker, similar to an advertisement. The marker is transmitted after a delay of 30 seconds. As 10 of the 100 web pages load very slowly over Tor and largely overlap with the marker, we increase the delay to 120 seconds in these cases.

The transmission time of the web page markers ranges from 12 to 20 seconds with a mean of 18 seconds. As the transmission happens in the background, these slight differences to the original web page traffic are hardly noticeable, in particular because communication over Tor often suffers from rather long loading times.

To enable a comparison with related website fingerprinting attacks, we implement the approaches by Herrmann et al. [4], Panchenko et al. [8] and Cai et al. [2], where the first is slightly modified to use an SVM instead of an MNB classifier. For all approaches, we conduct the same experimental procedure as for the Torben attack, except that we do not implant markers into the web pages. Furthermore, our detection method is trained solely on web page markers, whereas the fingerprinting attacks are trained on a sample of the web pages visited in February 2014.

Figure 2a shows the performance of Torben and the website fingerprinting attacks on the web pages visited in February 2014. Our approach is able to correctly deanonymize 95% of the web page visits, whereas the performance of the other attacks ranges from 38.9% to 90.6%. However, all website fingerprinting attacks suffer from changes in web content. When applied to the same web pages visited in April 2014, none of the passive approaches is able to correctly identify more than a third of the web pages, as shown in Figure 2b. By contrast, the Torben attacks attains a similar performance as in February 2014 and outperforms the website fingerprinting attacks, as the learned markers do not change over time.

Overall, this experiment demonstrates that the proposed side-channel attack can deanonymize web page visits with high accuracy in a closed-world setting.

4.4 Open-World Evaluation

In this experiment, we consider an open-world setting, where the user can freely visit web pages from a large unknown set and only few of these pages are tagged with a web page marker. The adversary is thus interested in determining whether marked web pages have been visited by a particular user.

We extend the previous experiment by additionally choosing 60,000 web pages at random from the Alexa top one million ranking, none of which are part of the top 100 web pages. Each of these web pages is then visited once over Tor without sending a marker. This large set of web pages enables us to estimate the false-positive rate of the Torben attack. Due to the usage of an SVM with probabilistic output, we can simply identify data transfer sequences that do not contain a web page marker by setting a threshold on the determined probabilities.

Figure 2c shows a ROC curve for the detection performance of Torben in this experiment. Note that the false-positive rate on the x-axis is given in the range 0 to 0.001, while the detection rate is shown on the y-axis between 0.6 and 1.0. The attack enables detecting more than 91% of the 100 web pages with no false positives in set of 60,000 additional web pages.

The outcome of the open-world evaluation demonstrates the reliability of the web page markers, which are unlikely to be confused with regular web page traffic and enable detecting marked pages with a very low false-positive rate. This reliability rests on the design of the side channel that makes use of atypical request-response pairs for transmitting information (Section 3.2).

4.5 Live Evaluation

The previous experiments already show the effectiveness of our approach on a large amount of data that has been recorded automatically. However, in order to fortify the obtained results, we conduct a further experiment in which we analyze web traffic of real users.

In this experiment, four different users surf through the web for roughly two hours using Tor. Each user visits arbitrary web pages and from time to time, randomly chosen marked web pages. We simulate the attack scenario of a remote marker as described in Section 4.3 and set the transmission delay for markers to 120 seconds.

Afterwards, the recorded traffic of each user is split into chunks of three minutes that are separately analyzed using a sliding window as described in Section 3.4. Hence, the classifier outputs a label and a probability score for each chunk. If the probability score of a chunk is below a particular threshold, we conclude that the user did not visit any marked web pages during this time frame. We select this threshold to be $t = 0.1$ based on several test runs which we conducted in advance. From a total of 34 visited marked web pages, we are able to detect and classify 31 correctly. Furthermore, the classifier does not output any false positives.

5. LIMITATIONS AND DEFENSES

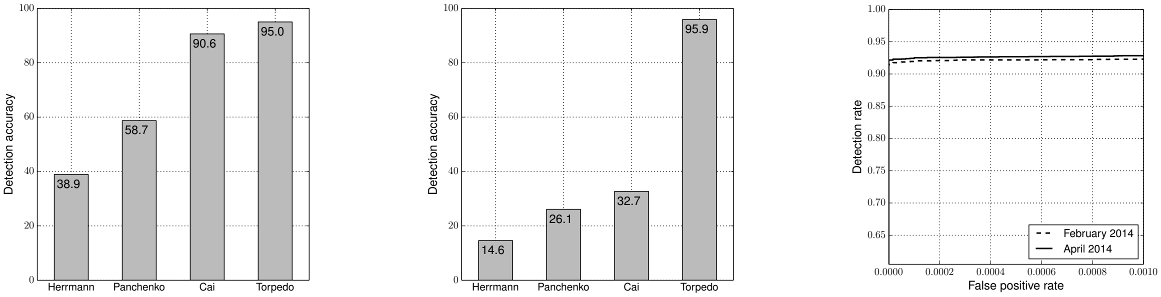
Our experiments show that Torben is a reliable deanonymization technique against Tor. However, its limitations could enable development of effective defenses.

First of all, our attack makes no attempts of hiding web page markers. In consequence, developing a detection mechanism that identifies and suppresses communication on our side channel may be an effective method to protect users. Such mechanisms could be implemented either on the client side directly in the form of a browser plug-in or even by Tor nodes to protect all clients at once.

Second, our evaluation shows that web page markers may be corrupted if they largely overlap with the loading of a web page. This suggests that, although our classification model is able to compensate some traffic interference using positional features, it fails if the interference becomes too severe. A second option would thus be to introduce chaff traffic to disturb the side-channel communication. Yet, this option may not be satisfactory in practice, as it introduces additional traffic and might lower Tor's overall performance.

6. RELATED WORK

Attacks against anonymization techniques have been a vivid area of research in the last years. Several researchers have recognized that, while it may not be possible to decipher encrypted messages, when transmitted over a communication link, important characteristics of the data such as its size and partitioning into data units is often disclosed. For example, Liberatore et al. have shown that it is pos-



(a) February 2014

(b) April 2014

(c) ROC curve

Figure 2: Evaluation of the detection performance in the closed-world (a,b) and open-world scenario (c).

sible to identify web pages in encrypted HTTP traffic by observing characteristic packet lengths [7].

Based on this observation, website fingerprinting attacks try to deanonymize Tor users by identifying patterns in the encrypted traffic between user and guard node, which are characteristic for particular monitored web pages [2, 4, 8]. All of these approaches, however, suffer from false-positive rates that are too high to enable practical application. Moreover, our experiment demonstrate that website fingerprinting suffers from changes in web content and degrades deanonymization performance over time.

While website fingerprinting attacks consider an attacker only between the client and its entry node, traffic confirmation attacks require an attacker who is able to correlate the traffic on both ends of a communication path. Several authors demonstrated the effectiveness of such attacks [6, 9]. However, this scenario requires the attacker to have access to both ends of the connection over a long period of time.

Passive traffic analysis attacks require traffic to be observed for a longer period of time before users can be effectively deanonymized. Approaches where attackers actively attack the communication path to lower the required amount of time have therefore been proposed. Several of these watermarking schemes have been proposed for mixed networks which use inter-packet delays to encode watermarking sequences [5, 10]. In contrast to Torben, these attacks do not take place at the application layer and are thus much more difficult to realize in practice. Furthermore, they consider a stronger attacker who is able to control the exit node or the link between the exit node and the server.

7. CONCLUSION

Tor is among the largest and best understood anonymization networks operated to date, protecting the privacy of over a million users worldwide. This paper presents a novel deanonymization attack on Tor that exploits a fundamental weaknesses of low-latency anonymization networks. In particular, we show that an attacker capable of providing web content to users, e.g., through banner advertisements or cross-site scripting, is able to deanonymize users via a side-channel attack. By transmitting web page markers through this side channel, the attacker can expose the web pages a user visits within a couple of seconds. This attack is considerably more effective than known website fingerprinting attacks and far less intrusive than browser exploits used in the wild. Fortunately, the side-channel communication is clearly visible in network traces and hence it may be possible to implement detection approaches as a first countermeasure against this attack.

Acknowledgements. The authors acknowledge funding from BMBF under the project INDI (FZK 16KIS0154K) and from DFG under the project DEVIL (RI 2469/1-1).

References

[1] D. Arp, F. Yamaguchi, and K. Rieck. Torben: Deanonymizing Tor communication using web page markers. Technical Report IFI-TB-2014-01, University of Göttingen, 2014.

[2] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2012.

[3] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. 2004.

[4] D. Herrmann, R. Wendolsky, and H. Federrath. Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier. In *Proc. of ACM Workshop on Cloud Computing Security*, 2009.

[5] A. Houmansadr and N. Borisov. Swirl: A scalable watermark to detect correlated network flows. In *Proc. of Network and Distributed System Security Symposium (NDSS)*, 2011.

[6] B. N. Levine, M. K. Reiter, C. Wang, and M. K. Wright. Timing attacks in low-latency mix systems. In *Financial Cryptography*, 2004.

[7] M. Liberatore and B. N. Levine. Inferring the source of encrypted http connections. In *Proc. of ACM Conference on Computer and Communications Security (CCS)*, CCS '06, 2006.

[8] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. Website fingerprinting in onion routing based anonymization networks. In *Proc. of ACM Workshop on Privacy in the Electronic Society*, 2011.

[9] V. Shmatikov and M.-H. Wang. Timing analysis in low-latency mix networks: attacks and defenses. 2006.

[10] X. Wang, S. Chen, and S. Jajodia. Network flow watermarking attack on low-latency anonymous communication systems. In *Proc. of IEEE Symposium on Security and Privacy (S&P)*, 2007.