# Abusers don’t get Privacy. Sensitively Logging and Blocking Tor Abuse

Matthias Marx[1]

**Abstract:** Tor has a significant problem with malicious traffic routed through Tor exit nodes. They create a credible reason for websites to discriminate against Tor users. The abuse also creates a strong disincentive to run exit nodes since the exit node operators have to deal with abuse messages and possible law enforcement interactions. We want to detect and mitigate the attacks that happen through Tor exit nodes without undermining Tor users’ anonymity and privacy. We use a modified version of the Tor exit node to enable NIDS (Network Intrusion Detection) monitoring and termination of malicious activity on a per-circuit level. We use the Zeek IDS (formerly Bro) to detect attacks using robust mechanisms that have very low false positive rates. Initial results indicate that, using our approach, the number of abuse cases can be reduced.

**Keywords:** Tor; Malicious Traffic; Intrusion Detection System

## 1 Introduction

Abusive use of Tor, where attackers take advantage of Tor’s anonymity to launch attacks, damages the Tor network in multiple ways. This includes acting as a significant disincentive to running a Tor exit, serving as a source of negative publicity, and creating a credible reason for websites to discriminate against Tor users.

Several publications describe the abusive use of Tor [Mc08, CMK10, Li15]. Over 20 % of the top Alexa websites discriminate against Tor users. Many of the website operators mention the network attacks passing through Tor as one of the main reasons for discriminating [Si17]. If we can prevent large scale attacks through Tor, we hope to remove major incentives for discriminating against Tor traffic.

The goal of this project is to develop and evaluate intrusion detection policies, guided by abuse complaints, which can run on a Tor exit node to mitigate the outbound attacks without disrupting normal user behavior and without posing a privacy risk to non-malicious users.

[1] Universität Hamburg, Sicherheit in verteilten Systemen, Vogt-Kölln-Straße 30, 22527 Hamburg, Deutschland
marx@informatik.uni-hamburg.de

doi:10.18420/sicherheit2020_16

## 2 Detection Schemes

Our detection schemes treat each Tor circuit as a separate user which is then evaluated with the Zeek[2] network intrusion detection system (NIDS). We modified the Tor exit source code to report to the NIDS the circuit ID associated with each TCP connection 4-tuple and each DNS lookup. Using this, the NIDS can evaluate each Tor circuit for abuse independently of all other circuits. This also enables the NIDS to optionally terminate a misbehaving circuit without affecting any other user.

We use Zeek with all default logging disabled. Only the limited logging (discussed below) occurs. We use a combination of existing Zeek rules and new rules to detect abuse. Our initial starting points for detecting and blocking abuse are:

1. **Port Scanning:** We use Threshold Random Walk (TRW) [Ju04] to detect port scanning both across hosts and within a single host.
2. **Brute Force Guessing:** We detect brute-force password guessing for HTTPS and SSH using TRW, with a short terminated connection (for SSH) and either a short terminated connection or a regular period of same-sized requests with short responses (for HTTPS) as indicative of failures for the TRW algorithm.
3. **Pattern Matching for HTTP Abuse:** We use regular expressions against the HTTP GET and POST requests which detect various attacks, such as SQL injection or cross site scripting.

It is critical that we use detectors with a very low false positive rate. TRW-type detectors are already well understood and have very low false positive while still being sensitive, and we evaluate the pattern matching for the HTTP abuse against known benign network traffic before deployment on our exit nodes. We will add subsequent detection routines as they prove useful, in particular in response to any abuse complaints we receive directly or through public IP abuse databases.

Upon detecting an attack, the NIDS enables logging for the alerted circuit and optionally terminates the Tor circuit which triggered detection to prevent further abuse after having identified this as a problematic circuit.

## 3 Methodology

We will operate two exit nodes running our IDS policies which analyze for abuse on a per-circuit basis: one which only logs alerts, and one which both logs alerts and terminates offending circuits. This logging only occurs for circuits which trigger our abuse detectors. For both we will evaluate the complaints received, and see if there is a substantial difference between the one which terminates offending circuits and the one which only logs.

[2] https://www.zeek.org/

**Data Collection** Intrusion Detection Systems normally collect a large amount of information. We perform just minimal logging for all circuits: the number of distinct hosts and the number of bytes transmitted, with both values truncated and the associated time intervals truncated. This is necessary to estimate the normal, non-malicious usage of the exit.

We only perform more detailed logs on circuits detected as malicious. Such logging includes all detected NIDS events associated with the circuit, in order. The NIDS events are in order but won’t be timestamped, and circuit creation is only recorded truncated to the nearest hour. Such logging only takes place once an hour, with the logged circuits presented in randomized order, so as to ensure no correlation between them.

The logs are essential for evaluation: they enable us to determine if our detectors effectively detect abuse (by correlating abuse complaints to log entries) and if such detection is sufficiently early to prevent abuse complaints. If we can’t correlate a complaint to a log entry, this suggests holes in our detection strategy.

**Modifications to Tor** We modified the Tor source code to generate new control events for RELAY RESOLVE and RELAY BEGIN requests which report the circuit ID and the requests’ contents. For RELAY RESOLVE cells this reports both the hostname and the answer (IP address, hostname or error). For RELAY BEGIN cells we report the TCP 4-tuple.

Using Stem[3], the NIDS subscribes to a stream of RELAY RESOLVE and RELAY BEGIN events and uses these events to associate the Tor circuit ID with the network traffic to determine attacks. Furthermore, the NIDS can instruct Tor to close a circuit via Stem.

**Privacy Concerns** The detection algorithms we are planning to use have very low false positive rates. This will be confirmed by experiments in a test network before experimenting on the live Tor network. We limit the granularity of data and will keep the logs private. We do not collect data that we do not need for attack detection. We use Zeek with all default logging disabled and make sure that Zeek does not contact any third party services during operation. Furthermore, we use offline relay identity keys, two factor authentication for SSH, unattended upgrades and limit the number of people who have access to server and data. We are discussing our approach with the Tor Research Safety Board.

Our detection does not require manual inspection of client traffic. The logs are essential for evaluation and do not contain enough information to deanonymize a circuit. They are not interconnected accross circuits and there is only the coarsest of timing information for circuit start and duration. Events within the circuit, although ordered, add very little timing information. An adversary cannot make the system log any other users’ traffic by inserting abusive circuits.

**Other Concerns** We plan to detect large scale attacks: port scanning, brute-force attacks and attacks with known signatures. Removing these abusive traffic will improve Tor’s image

[3] `https://stem.torproject.org/`

and will reduce websites preemptively blocking Tor. Additionally the blocking, by being selective, should not affect normal users.

As a result of our mitigation scheme, attackers could change their behavior. Abusive users could spread their streams over more circuits. We will investigate by how much attacks slow down when we distribute them over many circuits. Furthermore, we examine our approach with the assumption that in the future all relays would mitigate malicious traffic. Therefore, we do not consider that abusive traffic could shift to relays that don't implement mitigation schemes.

## 4 Evaluations

**Pattern Matching for HTTP Abuse** We use Zeek and the CICIDS2017 dataset [SLG18] to analyze labeled PCAPs that contain realistic background traffic and common attacks. For detection of SQL injection scanners, we use Zeek's standard module. Detection of cross site scripting scanners is based on NoScript's[4] regular expressions.

We vary the monitoring interval and the threshold that determines if an attack is ongoing based on the number of requests that appear to be suspicious. We want a high threshold to achieve a low false positive rate and at the same time we want a low threshold to detect attacks early.

Figure 1 shows the influence of the threshold parameter for cross site scripting attacks. The results show for a monitoring interval of 10 seconds that no false positives occur for a threshold of 11 or more requests. Attackers who do not want to be detected must use a new circuit every 10 requests. With a monitoring interval of 10 minutes, no false positives occur for a threshold of 25 requests or more.

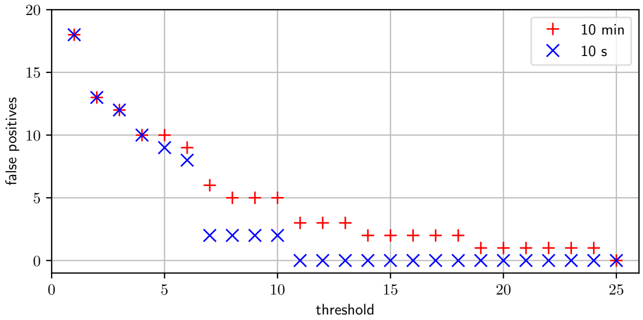

Fig. 1: Number of false positives of regular expressions used to detect cross site scripting attacks for different monitoring intervals (10 s, 10 min) and request thresholds.

[4] `https://noscript.net/`

Based on the results, in the following experiments we will choose a request threshold of 5 for SQL injection and a threshold of 25 for cross site scripting attacks. Later, we may change the thresholds based on findings from the abuse complaints.

**Spreading Attacks Across Circuits** We investigate if attacks could simply spread across many circuits to avoid detection. Figure 2 shows that it does not make much difference if one or ten circuits are used to request a website one hundred times. This means that we have to detect attacks after very few requests or circuit creation has to be made more expensive, for example through proof-of-work.

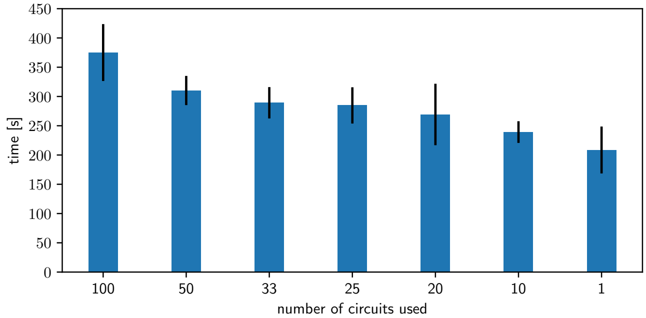

Fig. 2: Time that is needed to request a website 100 times over Tor using one or multiple circuits.

**Detecting Abuse on Emulated Tor Exit Nodes** We use chutney[5] for running multiple instances of Tor and NetMirage[6] to set up the emulated network. We use several Metasploit[7] modules to run the following attacks: **Port Scanning** across hosts and within a single host, **Brute Force Guessing** of HTTP Basic Auth and SSH credentials, and **SQL injection** and **cross site scripting scanning**. Our experiments in the lab show that the detectors work. We can detect and close circuits that are used for attacks.

**Mitigating Abuse on the Live Tor Network** We run two exit nodes with reduced exit policies that allow web browsing (ports 80 and 443), SSH (port 22) and port scanning to a limited extent. One exit will only log alerts, and the other will both log alerts and terminate offending circuits. For both we will evaluate the complaints received, and see if there is a substantial difference.

**Tor Network Telescope** We run a network telescope (analog to [Mo04]) to measure Tor's background noise. We monitor nine unused /8 IPv4 subnets that carry no legitimate traffic. Using ping-scans and BGP data, we have verified that the subnets are actually not used. In the logs we include the following information: circuit ID, number of distinct scanned addresses per /16 subnet, scanned ports and timestamp (hourly granularity). We omit logs for circuits with less than 25 connection attempts. The measured noise will be used to draw conclusions about (undirected) attacks carried out through Tor. On detection of port

[5] `https://gitweb.torproject.org/chutney.git/`

[6] `https://crysp.uwaterloo.ca/software/netmirage/`

[7] `https://www.metasploit.com/`

scanning, we will redirect traffic to a honeypot to provide actionable intelligence on how attackers are using Tor for exploitation.

## 5 Conclusion & Future Work

In this work, we proposed a new defense against abusive use of Tor. We modified Tor to enable monitoring and termination of malicious activity on a per-circuit level. Initial results indicate that, using our approach, attacks can be detected and prevented. It remains to be checked whether the number of abuse complaints can actually be reduced.

In the future, our approach could be extended to provide network-wide statistics on abuse of Tor. To further enhance privacy, the NIDS could run in an encrypted enclave that only outputs circuit IDs for circuits that are deemed malicious. Alternatives to termination of circuits, such as degradation of quality of service, should also be investigated.

## Acknowledgements

The author wants to thank Sadia Afroz for the fruitful discussion and the anonymous reviewers for their useful comments and suggestions. This work was partially supported by NSF grant CNS-1518918.

## Bibliography

[CMK10] Chaabane, Abdelberi; Manils, Pere; Kaafar, Mohamed Ali: Digging into anonymous traffic: A deep analysis of the Tor anonymizing network. In: International Conference on Network and System Security. IEEE, pp. 167–174, 2010.

[Ju04] Jung, Jaeyeon; Paxson, Vern; Berger, Arthur W; Balakrishnan, Hari: Fast portscan detection using sequential hypothesis testing. In: Security and Privacy. IEEE, pp. 211–225, 2004.

[Li15] Ling, Zhen; Luo, Junzhou; Wu, Kui; Yu, Wei; Fu, Xinwen: TorWard: Discovery, Blocking, and Traceback of Malicious Traffic Over Tor. IEEE Transactions on Information Forensics and Security, 10(12):2515–2530, 2015.

[Mc08] McCoy, Damon; Bauer, Kevin; Grunwald, Dirk; Kohno, Tadayoshi; Sicker, Douglas: Shining light in dark places: Understanding the Tor network. In: Privacy Enhancing Technologies Symposium. Springer, pp. 63–76, 2008.

[Mo04] Moore, David; Shannon, Colleen; Voelker, Geoffrey; Savage, Stefan et al.: Network telescopes. Technical report, Cooperative Association for Internet Data Analysis, 2004.

[Si17] Singh, Rachee; Nithyanand, Rishab; Afroz, Sadia; Pearce, Paul; Tschantz, Michael Carl; Gill, Phillipa; Paxson, Vern: Characterizing the Nature and Dynamics of Tor Exit Blocking. In: USENIX Security. 2017.

[SLG18] Sharafaldin, Iman; Lashkari, Arash Habibi; Ghorbani, Ali A: Toward: Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In: ICISSP. pp. 108–116, 2018.