Timing Analysis Attack Simulation in Tor Networks

Matteo Martelli, Davide Berardi

June 4, 2015

Abstract

In this document we talk about blablabla

1 Introduction

Protecting data privacy on the Web is a very hot topic nowadays. Users of the Web may want to surf without the risk that their personal informations can be read by other users. One of the most largely-used architecture for this purpose is the *Onion Routing* and its protocol implementation: *Tor*[2]. In fact, the latter is modeled with several techniques with the aim to provide communication security and data privacy to its network users. Anyway there have been recent papers pointing out the Tor's vulnerabilities. As the Tor community itself stages, "Tor does not provide protection against end-to-end timing attacks"[1], thus the chance for an attacker to eavesdrop a Tor communication traffic and discover the users involved in it by a timing analysis is a well known vulnerability of Tor.

In this kind of analysis, in order to identify the source of a communication, the attacker should be able to trace the outgoing traffic and the incoming traffic from both the entering and exiting node of the communication path. Clearly a timing analysis is feasible only under a certain amount of conditions that are often hard to satisfy. As instance, discovering that a generic user U is connecting to a server S over a Tor communication may require tracing the traffic of many users in the network, as the attacker cannot know which users may be interested in connecting with S. Also, there may be the need of tracing more than just the interested server because the attacker can find a better time relation between the user U and another server S' than between U and the interested server S, thus the attacker could exclude U to be a possible connection source for S. In the section 2 we will better describe how Tor works and how the end-to-end timing attack could be performed.

In order to test the feasibility and the parameters involved in a time analysis attack over the Tor network, we set up a simulation scenario in which a series of simulation runs have been performed and some interesting empirical results have been taken out and analyzed. At the end we will point out how the Tor time analysis vulnerability can be critical and we will introduce some proposals to enhance Tor with the view of preventing this kind of attacks.

2 Tor

Tor is an implementation of the onion routing architecture model. The onion routing consists in a technique that provides anonymous connections over a computer network[3].

Tor is born with the aim to allow people to improve their privacy and security on the Internet. Its architecture is based on the Onion Routing model and it's widely used by many user over the world. Users may be interested on using Tor for different purposes such as avoiding website tracking, communicating securely over Internet messaging services, or just web surfing with the access on the services blocked by their local Internet providers.

The idea behind Tor, and Onion Routing as well, is to protect people against a common form of Internet surveillance known as "traffic analysis". Traffic analysis can reveal information about the network traffic such as the source, destination, size, timing and more of the analyzed traffic packets. This can be possible even if the packets are cyphered because the traffic analysis focuses on the header part of the packets that are in plain text.

Thus, simply listing between the sender and recipient on the network, a traffic analysis can be performed. Moreover, spying on multiple parts of the Internet and using some statistical techniques, some attackers can track the communications patterns of many different organizations and individuals.

In the next paragraphs we will discuss more in details about Tor communications and the flaws of the model.

2.1 Tor Internals

The figure 1 shows how a message is cyphered before the communication begins. The communication source, before sending the message, choses a communication path of nodes which the keys are known to the sender. Then the source node is able to create a stack of encryption starting with the key of the last relay node and then continuing backwards with the keys of the other relay nodes in the chain. In this way every node in the communication path can decrypt the package and read the next hop address. After that the final node receives the message, he can send the response back to the originator of the data stream. In this phase the response message is encrypted sequentially by each node in the chain. With this method each relay node can gain access to the previous and the next node addresses only. Anyway the last node of the Onion Routing path, called exit node, send the message to the end point as plain text.

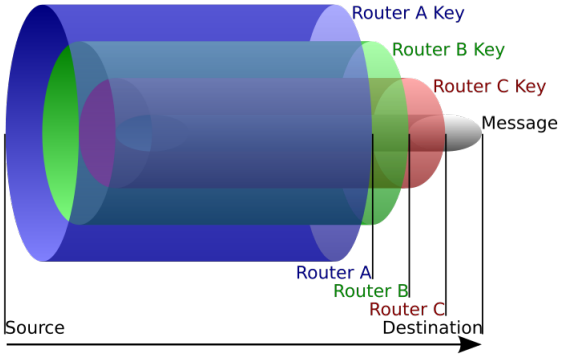

Figure 1: Message encryption layers.

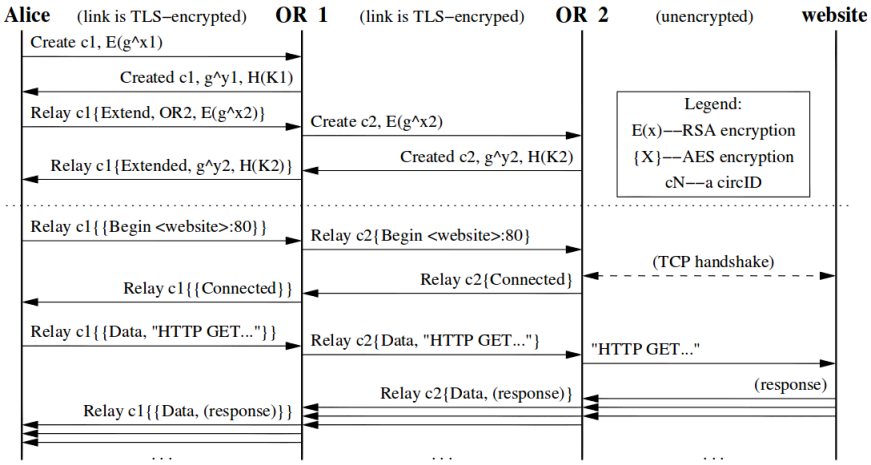

Figure 2: A two relay Onion Router communication.[2]

2.2 Tor attacks

Recent studies showed the existence of different kinds of possible attacks on the Tor network. Apart from Side channel attacks [1] we can classify these attacks in two major families: probabilistic attacks and path selection forcing attacks. The first ones are based on the data analysis that leak from the network. In these cases an attacker that sniff some specific data packets can perform some statistical analysis and may be aware of users identity informations. On the other hand, the path selection forcing attacks are based on some techniques able to route the network traffic through a sequence of nodes handled by the attacker. In our study case we focus on one of the probabilistic attacks: Timing Analysis Attack.

2.2.1 Timing Attack

As Tor development team states Tor is vulnerable to the timing attack analysis:

> *"Tor does not provide protection against end-to-end timing attacks[...]"*

More specifically, an attacker placing between the communication source and the Tor entry node and also between the Tor exit node and the communication destination, can observe the connection time of both end-points and point out their relation. The attacker can, for example, inject some malware code in the communication source machine or in some other entities, i.e. a router, on the network segment before the entry node. Similarly, a way to introduce some eavesdropping point into the communication destination could be found by the attacker.

[1] For example the well known *tor browser attack* .

We will focus on how much this kind of attack can be effective on a real network, considering the amount of the resources available to the attacker.

3 Simulation

3.1 Shadow

3.2 Autosys plugin

3.3 Analyzer plugin

3.4 Simpletcp plugin

4 Data Analysis

4.1 Netbuilder Script

4.2 Analyzer Script

4.3 Empirical Results

5 Future Works

6 Conclusions

References

[1] *Tor: Overview.* `https://www.torproject.org/about/overview.html.en`.

[2] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[3] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing. *Communications of the ACM*, 42(2):39–41, 1999.