Creating a Simulated Dark Web Environment

Mohamed Hossam Queena

March 12, 2020

Contents

1 Introduction

2 The onion routing

2.1 2.1 TOR

2.1.1 2.1.1 What is Tor?

The onion routing (TOR) is a network that is implemented to anonymize TCP requests from applications like web browsers, instant messaging[8]. TOR is, without a doubt, one of the most extensively used technology for safe and anonymous web usage[1], with more than 2 million users who use it on daily basis[4]. It is also used to avoid censorship policies in some jurisdictions, and be able to have free browsing[3].At the same time, TOR is an expanding research network for researching who experiment to enhance to the network resistance to attacks, and performance[3]. The implementation of TOR secures the communication between the source and the destination, putting in mind that the entry and exit relays are not compromised by the attacker at the same time[4]. These relays are distributed all over the world through volunteers[5]. TOR's aim to producing low-latency anonymity to it's users, has generated a lot of research topics, regarding not only anonymity attacks and defences, but also performance and scalability improvements[5].

2.1.2 2.1.2 How does TOR work?

TOR is a three-layered anonymous network, where each layer knows data about the next hop in the circuit of the network[2]. Let's start by explaining simply: when you send a request for any kind of data, TOR forwards your request to another computer in the TOR network, then again this computer, in it's turn, forwards your request to another computer, and then another, and then finally the third computer in the network sends the data to the original destination, acquiring the desired data, and sending it back to the user through another path[6]. Now let's go into details, clients using the TOR network send their TCP connections through a small route of TOR routers, a normal route would consist of: an entry guard, a middle router, exit router[3]. Every router in the network maintains a TLS connection with every other router in the onion network[8]. A crucial note is that only the entry node directly knows the original source of the request being sent, and only the exit node can directly know the destination of the request being sent, and also take a look at the decrypted request payload[7].

2.2 2.2 Simulating TOR

There has been a lot of techniques to test TOR's performance, and these techniques included theoretical modeling, private TOR networks, distributed overlay network deployment, simulation and finally emulation[4]. Simulation is very good in scalability, it does not go without decreasing the accuracy of the performance, as simulators are becoming outdated due to the ongoing updates of TOR[5].

2.2.1 2.2.1 Previous experiments

There has been previous experiments and trials to simulate TOR, and we are going to list some of the noticeable works that has been done. Let's start by Shadow, their simulation runs on one machine, and that's why it's being called TOR in a Box. They take control of all details and aspects of the experiment they are running, while running the real TOR software[5]. Although Shadow runs on a single machine, it is able to run a huge private network[5]. Shadow is friendly with average hardware hosts[2]. Shadow is used to investigate TOR's network congestion and performance problems, in order to try and improve TOR's client performance[5]. Shadow allows you to model the TOR network as you wish; meaning that you can have different numbers of clients and relays in your private network. Shadow is based on virtualization obviously, so it uses something called virtual nodes, where each virtual node is basically a simulated host. In Shadow's virtual network, each virtual node is, typically, assigned an IP address to be able to communicate with other nodes[5]. A sample of a configuration for Shadow is 10r40c; meaning that you have ten relays and 40 clients, and there many different combinations like this one, so you can test the network's performance on different sizes[2]. We are more interested in the client's performance while using tor, so in Shadow they measure the client's time to receive the first byte, and the time to complete a download. They found out that, as expected, TOR's download time is much higher than the typical direct download time[2]. Shadow's results also proves that higher number of relays decreases the transmission time[5]. Leaving Shadow and moving on to ExperimenTor, like Shadow they use only one machine to simulate the whole experiment, however they specify that a Linux 2.6.32 machine was used[3].

2.2.2 2.2.2 Why is simulating TOR better than emulating TOR?

References

[1] Michael Backes, Ian Goldberg, Aniket Kate, Esfandiar Mohammadi, *Provably Secure and Practical Onion Routing*. 2012.

[2] Hartanto Kusuma Wardana, Liauw Frediczen Handianto, Banu Wirawan Yohanes, *The Onion Routing Performance using Shadow-plugin-TOR*. 2017.

[3] Kevin Bauer, Micah Sherr, Damon McCoy,Dirk Grunwald, *ExperimenTor: A Testbed for Safe and Realistic Tor Experimentation*. 2011.

[4] Fatemeh Shirazi, Matthias Goehring, Claudia Diaz, *Tor Experimentation Tools*. 2015.

[5] Rob Jansen, Nicholas Hopper, *Shadow: Running Tor in a Box for Accurate and Efficient Experimentation*. 2012.

[6] Eric Jardine, *The Dark Web Dilemma: Tor, Anonymity and Online Policing*. 2015.

[7] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, Douglas Sicker, *Shining Light in Dark Places: Understanding the Tor Network*. 2008.

[8] Roger Dingledine, Nick Mathewson, Paul Syverson, *Tor: The Second-Generation Onion Router*. 2008.