Trust but Verify: Analysis of Flawed Trust Beliefs for Tor and a Passive Routing Attack

Nicholas Schmeller

In a Tortoiseshell: *In a paper that exposes a potential weakness of Tor, a supposedly secure and private internet browsing protocol, Nicholas Schmeller positions nontechnical readers to appreciate the significance of his work by effectively presenting* ***motive****. By presenting the necessary technical jargon in an easily digestible manner in combination with the inclusion of practical illustrations, Nicholas ensures that all readers can grasp the complication resulting from the fact that, although an agent's information is secure while in transit under the Tor protocol, said information is vulnerable at the very beginning and end of its virtual journey.*

Excerpt

Secrecy and anonymous communications have been essential to every human society that has ever existed. Even before the age of near-instant communication, people have sought to conceal their messages for their own purposes, sometimes good and sometimes bad. Concealment was accomplished by two general techniques: making the message hard to find (steganography) and making the message hard to understand (cryptography). Secret communications have grown to improve at both of these broad techniques, and indeed modern cryptography has reached near-unbreakable strength with the introduction of public-key encryption methods such as RSA.

Encryption, however, is only secure when the message is actually encrypted, leaving vulnerabilities at the endpoints before encryption and after decryption. Additionally, while an adversary may not be able to understand a message, they can still determine where the user sent the message. In an extreme example, a political dissident living in an oppressive state would not wish to be caught sending anything to a free-speech advocacy group, regardless of whether the message is able to be read. Virtual Private Networks (VPNs) are a basic protection against this type of surveillance, but determined adversaries can potentially compromise the VPN node, and powerful adversaries such as a state can even block internet access to the VPN. Clearly, a solution that decentralizes communications, employing steganography, is most desirable.

The Onion Router project, commonly known as The Tor Project or simply Tor, is one particularly effective result of the effort to perfect usage of both cryptography and steganography. The basic concept of Tor is to send user traffic through multiple *relays*, forming a simple graph, and to use multiple layers of encryption to make the information passed along each edge strongly

and equally secure. Tor compares user traffic to an onion: a message is sent from the user with multiple layers of encryption, and each relay along the path to its destination is only able to peel off one layer of encryption. This ensures that, at any given moment along the transmission process, the information is only readable by the relay that sent it and the relay that will receive it.

While this is a strong model of anonymous communications, vulnerabilities exist at the endpoints of the path between user and destination. Even if it is encrypted and thus unable to be read by an adversarial listener between the user and the first relay, the information that a user sends to the initial Tor relay can be timestamped. Likewise, a listener between the final Tor relay and the destination, even if it is encrypted and its source is hidden, can also be timestamped. A clever adversary can therefore *correlate* these timestamps to pair a user's location with her information's destination [1][2]. Preventing this correlation is an active research topic in the Tor community and studying the steganographic strength of Tor is the ultimate goal of this paper.

Author Commentary

Nicholas Schmeller

Computer science seems a lot more complicated than it really is. That's not to say that it is not difficult, or time-consuming, or soul-draining, because it can be all of those things. At the same time, however, what computer science has going for it that makes it such a popular and powerful discipline is the concept of *abstraction*: doing something really, really well and then hiding that thing behind a clean and simple layer of separation to make it easy to use for others. A great example of abstraction is what your internet browser did to get this commentary to you: all you had to do was click on a Writing Center link or a Google search result, and you arrived here, but a truly amazing eighth-wonder-of-the-world Rube Goldberg-esque process happened in the background. Abstraction makes ideas in computer science straightforward to understand, even if the abstracted concepts are convoluted and nit-picky. Abstraction is so great that I like to think the reason that I discovered my passion for computer science is because the COS 126 lecture-style was abstracted.

What does abstraction have to do with a paper about Tor? Computer scientists often use abstraction in day-to-day life to calibrate just how much detail they should reveal when they're explaining their projects to people who don't want a technical report—almost everyone. This is also called putting things into layman's terms, and in fact using a simple vocabulary to communicate an idea is a powerful form of abstraction. I got a lot of practice abstracting my Tor research over several months of explaining it to friends and family, and when it came time to write my paper, I used the examples that worked best with them (political dissidents and the onion metaphor) to decrypt what exactly I was doing.

So how do layers of abstraction manifest themselves in my introduction? My first paragraph is an extremely top-level description of information security (after all, most paragraphs that use the word "society" aren't going to go into a lot of detail). My second paragraph pierces the broadest layer of abstraction to explain a flaw of encryption. Note that I could have talked about other things in this second layer, for example what types of internet traffic use encryption, how encryption methods are implemented, etc., so while I am going into more detail, I'm also beginning to direct focus to my topic. My third paragraph and layer explains a good solution for the flaw in the second paragraph (specifically, Tor). Again, note that I could have brought up other solutions that exist in this layer, and that I actually gave a strawman solution in my second paragraph to give perspective to why I'm diving deeper into these layers of abstraction. My fourth

and final paragraph pierces the layer of Tor to go into some details of Tor implementation and their flaws, and then we end up right where we need to be for me to start explaining my research.

I used the example of abstraction in this commentary because that is the lens with which I viewed my own writing process. I knew that information security was an intimidating topic for a lot of people, so I felt a heavier onus to make my work understandable and relevant. If you take a step back and look at what I was doing, though, you'll probably see a structure that's not too different from your writing seminar discussions about motive.

.

Editor Commentary

Nicholas Johnson

Despite the emphasis placed on the notion of motive in Princeton's Writing Seminars, most seminars unfortunately do not directly touch on or demonstrate what motive can look like in papers from science, technology, engineering, and mathematics (STEM) disciplines. Accordingly, in my experience conferencing with students as a Writing Center Fellow, a recurring point of confusion when working with students producing STEM papers is how they can map the concept of motive from their Writing Seminar onto their technical work. Nicholas's piece is particularly effective in demonstrating what motive can look like in a STEM paper because he successfully provides a clear and compelling articulation of the significance of his work and presents it in a manner that is comprehensible to a reader lacking technical domain knowledge.

The working definition of motive adopted by the Princeton Writing Program follows Gordon Harvey's definition of the term as the intellectual context provided near the beginning of the paper to establish the significance of the author's contribution to the existing scholarly conversation. In essence, motive can be thought of as both a positioning of an author's thesis with respect to the current state of a discussion about an issue and an articulation of its importance. Within this framework, the technical jargon inherent to many STEM disciplines complicates the presentation of motive because often the true nature of an author's contribution can only be appreciated by a reader who has very strong technical domain knowledge. This introduces the following dilemma should the author wish his work to be understandable by a broad audience: that of defining all of the technical concepts necessary to understand the author's contribution before presenting motive and risk the reader figuratively getting lost in the details, or present motive earlier and risk the reader not being able to appreciate the true significance of the author's contribution.

Nicholas's introduction is particularly effective because he is able to strike a near ideal balance between the two aforementioned extremes. As noted in his commentary, Nicholas adopts a stepwise approach in his introduction, offering more in-depth contextual information at each step that ultimately leads to the presentation of his motive. Although the motive only appears at the end of his final introductory paragraph, all that comes before is critical for a typical reader to be able to grasp the essence of his motive. At each stage of his introduction, Nicholas presents increasingly specific technical domain concepts while explaining them in a manner easily digestible by nontechnical readers, in part through his emphasis on presenting use cases or challenges that a typical reader would be able to appreciate. His inclusion of the example

regarding a hypothetical political dissident, although extreme, is particularly effective in offering a very concrete, understandable, and practical illustration of the importance of his topic. All the while, Nicholas is careful to only present the technical concepts that are critical to understanding his motive in this section and leaves further technical but slightly less critical orienting material to be presented later in his work.

Works Cited

[1] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, “Users get routed: Traffic correlation on tor by realistic adversaries,” ACM SIGSAC, 2013.

[2] M. Nasr, A. Bahramali, and A. Houmansadr, “DeepCorr: Strong flow correlation attacks on Tor using deep learning,” arXiv, 2018.

Bios

Nicholas Schmeller ’21 is a computer science major from Medina, Ohio. Outside of his classes, he works on miscellaneous side coding projects, plays violin in the university orchestra, and is an orientation leader for Outdoor Action. He wrote this as a sophomore.

Nicholas Johnson ‘20 is a rising senior in the Operations Research and Financial Engineering Department, focusing on issues of privacy and fairness in machine learning. He is pursuing minors in Applications of Computing, Applied and Computational Mathematics, and Statistics and Machine Learning. His hometown is Montréal, Québec. Nicholas is a Residential College Advisor in Whitman, is the President of Princeton’s Chapter of Tau Beta Pi, and works as a Writing Center Fellow in the Writing Center. During his free time, he loves to play basketball, work out, and play chess.