"One cell is enough to break Tor's anonymity"

Posted February 19th, 2009 by arma

Tomorrow there's a talk at Black Hat DC by Xinwen Fu on an active attack that can allow traffic confirmation in Tor. He calls it a "replay attack (http://www.cs.uml.edu/~xinwenfu/paper/ICC08_Fu.pdf) ", whereas we called it a "tagging attack" in the original Tor paper, but let's look at how it actually works.

First, remember the basics of how Tor provides anonymity. Tor clients route their traffic (https://www.torproject.org/images/htw2.png) over several (usually three (https://wiki.torproject.org/noreply/TheOnionRouter/TorFAQ#VariablePathLength)) relays, with the goal that no single relay gets to learn both where the user is (call her Alice) and what site she's reaching (call it Bob).

The Tor design doesn't try to protect (https://www.torproject.org/svn/trunk/doc/design-paper/tor-design.html#subsec:threat-model) against an attacker who can see or measure both traffic going into the Tor network and also traffic coming out of the Tor network. That's because if you can see both flows, some simple statistics (http://freehaven.net/anonbib/#danezis:pet2004) let you decide whether they match up.

Because we aim to let people browse the web, we can't afford the extra overhead and hours of additional delay that are used in high-latency mix networks like Mixmaster (http://freehaven.net/anonbib/#mixmaster-spec) or Mixminion (http://freehaven.net/anonbib/#minion-design) to slow this attack. That's why Tor's security is all about trying to decrease the chances that an adversary will end up in the right positions to see the traffic flows.

The way we generally explain it is that Tor tries to protect against traffic analysis, where an attacker tries to learn whom to investigate, but Tor can't protect against traffic confirmation (also known as end-to-end correlation), where an attacker tries to confirm a hypothesis by monitoring the right locations in the network and then doing the math.

And the math is really effective. There are simple packet counting attacks (Passive Attack Analysis for Connection-Based Anonymity Systems (http://freehaven.net/anonbib/#SS03)) and moving window averages (Timing Attacks in Low-Latency Mix-Based Systems (http://freehaven.net/anonbib/#timing-fc2004)), but the more recent stuff is downright scary (http://www.lightbluetouchpaper.org/2007/05/28/sampled-traffic-analysis-by-internet-exchange-level-adversaries/) , like Steven Murdoch's PET 2007 paper about achieving high confidence in a correlation attack despite seeing only 1 in 2000 packets on each side (Sampled Traffic Analysis by Internet-Exchange-Level Adversaries (http://freehaven.net/anonbib/#murdoch-pet2007)).

What Fu is presenting in his talk is another instance of the confirmation attack, called the tagging attack. The basic idea is that an adversary who controls both the first (entry) and last (exit) relay that Alice picks can modify the data flow at one end of the circuit ("tag" it), and detect that modification at the other end — thus bridging the circuit and confirming that it really is Alice talking to Bob. This attack has some limitations compared to the above attacks. First, it involves modifying data, which in most cases will break the connection; so there's a lot more risk that he'll be noticed. Second, the attack relies on the adversary actually controlling both relays. The passive variants can be performed by an observer like an ISP or a telco.

Tagging attacks on designs like Tor go back over a decade in the literature. In the 1996 Onion Routing paper (Hiding Routing Information (http://freehaven.net/anonbib/#onion-routing:ih96)), they wrote:

> *If the responder's proxy [exit node] is compromised, and can determine when the unencrypted*

> *data stream has been corrupted, it is possible for compromised nodes earlier in the virtual circuit to corrupt the stream and ask which responder's proxy received uncorrupted data. By working with compromised nodes around a suspected initiator's proxy [Tor client], one can identify the beginning of the virtual circuit.*

When we designed Tor, we made a conscious decision to not design against this attack, because the other confirmation attacks are just as effective and we have no fix for them, and because countering the tagging attack doesn't come for free. Here's a quote from the Tor design paper (https://www.torproject.org/svn/trunk/doc/design-paper/tor-design.html#subsec:integrity-checking) in 2004:

> *Because Tor uses TLS on its links, external adversaries cannot modify data. Addressing the insider malleability attack, however, is more complex.*
>
> *We could do integrity checking of the relay cells at each hop, either by including hashes or by using an authenticating cipher mode like EAX, but there are some problems. First, these approaches impose a message-expansion overhead at each hop, and so we would have to either leak the path length or waste bytes by padding to a maximum path length. Second, these solutions can only verify traffic coming from Alice: ORs would not be able to produce suitable hashes for the intermediate hops, since the ORs on a circuit do not know the other ORs' session keys. Third, we have already accepted that our design is vulnerable to end-to-end timing attacks; so tagging attacks performed within the circuit provide no additional information to the attacker. Thus, we check integrity only at the edges of each stream.*

Basically, we chose a design where the tagging attack is no more effective than the passive confirmation attacks, and figured that would have to be good enough.

I should also note here that the correlation attack papers mostly focus on _observing_ flows at either end and matching them up. They do great with just that. But think what you could do if you actually control one of the endpoints, and can modulate how much you send, when you send it, etc. Heck, if you're the exit relay, you can spoof a much larger webpage for the user, and then have even more bytes to work with. None of that requires a tagging attack.

So, where does the "one cell is enough" come in? If you can tag a single cell and recognize it on the other end of the circuit, then you can be confident you've got the right flow.

One of the unknowns in the research world is exactly how quickly the timing attack succeeds. How many seconds of traffic (and/or packets) do you need to achieve a certain level of confidence? I'll grant that if you run the entry and exit, tagging is a very simple attack to carry out both conceptually and in practice. But I think Fu underestimates how simple the timing attack can be also. That's probably the fundamental disagreement here.

For example, Bauer et al. in WPES 2007 described a passive timing attack on Tor circuit creation that works before a single relay cell has been transmitted (Low-Resource Routing Attacks Against Tor (http://freehaven.net/anonbib/#bauer:wpes2007)).

Fu's paper also cites concern about false positives -- it seems on first glance that a tagging attack should provide much higher confidence than a timing attack, since you'll always be wondering whether the timing attack really matched up the right circuits. Here's a quote from one of the authors of Locating Hidden Servers (http://freehaven.net/anonbib/#hs-attack06) , another paper with a successful timing attack on Tor:

> *We were prepared to do parameter tuning, active timing signature insertion, etc. should it be necessary, but it wasn't. In the thousands of circuits we ran we _never_ had a false positive. The Bauer et al. paper from WPES'07 extended our work from hidden server circuits to general Tor circuits (although it only ran on a Tor testbed on PlanetLab rather than the public Tor network). The highest false positive rate they got was .0006. This is just a nonissue.*

One of the challenges here is that anonymity designs live on the edge. While crypto algorithms aim to be so good that even really powerful attackers still can't succeed, anonymity networks are constantly balancing performance and efficiency against attacks like these. One of the tradeoffs we made in the Tor design is that we accepted end-to-end correlation as an attack that is too expensive

to solve, and we're building the best design we can based on that.

If somebody can show us that tagging attacks are actually much more effective than their passive timing counterparts, we should work harder to fix them. If somebody can come up with a cheap way to make them harder, we're all ears. But while they remain on par with passive attacks and remain expensive to fix, then it doesn't seem like a good move to slow down Tor even more without actually making it safer.

Overall, I'm confused why the conference organizers thought this would be a good topic for a Black Hat talk. It's not like there's a shortage of "oh crap" Tor attack papers lately. I guess my take-away message is that I need to introduce Jeff Moss to Nick Hopper et al from Minnesota (How Much Anonymity does Network Latency Leak? (http://freehaven.net/anonbib/#tissec-latency-leak)) and Sambuddho Chakravarty et al from Columbia (Approximating a Global Passive Adversary against Tor (http://www.google.com/search?q=columbia+global+passive+adversary+tor)).

On February 19th, 2009 Anonymous (not verified) said:

are saying fu just wants attention by using a 12 year old attack he hoped no one at the conference would know about?

and that tor is secure for some narrowly defined use case?

i am seeing that attacking tor no matter how trivial is a great way to generate press for yourself. fu and marlinspike are winning the attention economy.

On February 19th, 2009 Anonymous (not verified) said:

for what it's worth, here is a previous take\paper from Fu on the subject: http://www.cs.uml.edu/~xinwenfu/paper/ICC08_Fu.pdf (http://www.cs.uml.edu/~xinwenfu/paper/ICC08_Fu.pdf)

On February 19th, 2009 Anonymous (not verified) said:

http://www.theregister.co.uk/2009/02/19/ssl_busting_demo/ (http://www.theregister.co.uk/2009/02/19/ssl_busting_demo/)

[Marlinspike] ran SSLstrip on a server hosting a Tor anonymous browsing network. During a 24-hour period, he harvested 254 passwords from users visiting sites including Yahoo, Gmail, Ticketmaster, PayPal, and LinkedIn. The users were fooled even though SSLstrip wasn't using the proxy feature that tricks them into believing they were at a secure site. Sadly, the Tor users entered passwords even though the addresses in their address bars didn't display the crucial "https." (Marlinspike said he later disposed of all personally identifiable information).

On February 19th, 2009 Anonymous (not verified) said:

the internet is not secure, it never was. marlinspike chose tor because he did not want to leave his parent's basement. i could take sslstrip to any store that uses wifi and grab all of their data too. starbucks, mcdonalds, etc are all fine places to do this attack too. just because some pimply faced youth failed to think big doesn't mean tor is insecure. tor is secure, perhaps not as anonymous as it once was, but from this panera store, tor is working fine. damn kids and your hyperbole

On May 7th, 2009 Anonymous (not verified) said:

Really interesting. Thanks for sharing this.
p.s. fix your theme :) i use IE8beta1 and your theme is not compatibly to my IE. In firefox3.06 is ok.
--

Mark G. cailis (http://cailis.biz/)

On February 19th, 2009 Anonymous (not verified) said:

http://www.theregister.co.uk/2009/02/19/ssl_busting_demo/ (http://www.theregister.co.uk/2009/02/19/ssl_busting_demo/)

SSLstrip works on public Wi-Fi networks, onion-routing systems, and anywhere else a man-in-the-middle attack is practical. It converts pages that normally would be protected by the secure sockets layer protocol into their unencrypted versions. It does this while continuing to fool both the website and the user into believing the security measure is still in place.

The presentation by a conference attendee who goes by the name Moxie Marlinspike is the latest demonstration of weaknesses in SSL

On February 19th, 2009 Anonymous (not verified) said:

http://www.informationweek.com/news/security/vulnerabilities/showArticle...
(http://www.informationweek.com/news/security/vulnerabilities/showArticle.jhtml?articleID=214501930&subSection=News)

Black Hat: Security Pro Shows How To Bypass SSL

Moxie Marlinspike (http://www.thoughtcrime.org/) captured 117 e-mail accounts, 16 credit card numbers, seven PayPal logins, and some 300 other miscellaneous secure login sessions in only 24 hours.

Marlinspike explained that he obtained such data by placing proxy software he'd written, called "sslstrip," on a node of a Tor network, to conduct what's known as a man-in-the-middle attack.

The proxy software intercepts HTTPS traffic, generates and signs security certificates, and mediates data passing between the client and server, capturing everything in the process. And though there are ways to detect the attack, like recognizing that a Web URL begins with HTTP rather than HTTPS, none of the test victims noticed.

The attack can also be augmented with the addition of a lock icon, which would suggest to most users that the session is secure, even if it's not.

Such tendencies allow Marlinspike to bypass SSL entirely. "Lots of times the security of HTTPS comes down to the security of HTTP, and HTTP is not secure," he explains in his presentation slides (https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf) .

Marlinspike plans to release sslstrip later this week.

On February 24th, 2009 Torrrr (not verified) said:

With regard to vulnerabilities - they are everywhere. People have to be careful about what they are doing - or they can expect trouble. The moral from this kind of stories is that we need some guidance on how to use Tor and what problems we can expect. Something like "Best Practises Guide" ?

Finally, I'd like to express my gratitude to the maintainers of the Tor project. Keep up the good work.

On February 27th, 2009 phobos said:

A "best practices for using Tor" guide has been on our todo list for ever, it's even on the volunteer page.

On February 25th, 2009 Anonymous (not verified) said:

Given, Tor is a subset of the internet points, Tor is easy to statistically manipulate begin & endpoint attacks, and there is a known number of jumps, Could Tor be statistically manipulate in

middle relay attacks by controlling a statistically significant number of relays to determine begin & endpoints? Statistically controlling enough Tor relays is the same as controlling all Tor relays, which is the same as controlling the begin and end points.

Many solutions are possible as one is trying to obscure a real identity for themselves. In the process of doing such, they have created one. Would it be better to get Tor into default settings on routers - OEM and third party (dd-wrt).

Other Possible Solutions
Package requests with fictitious requests. (Which is the real one?)
Pass data back that was not requested with data that was. (Which is real one?)
Combine two and cache requests for later retrieval of data. End user client will know which data is theirs, but multiple people will have requested such data on the network at similar times, making it appear more viral than individual. - Issue - others see random real and fictional requests on network and bandwidth increases.
Integrate Tor into Router Firmware to increase number of Tor relays. (think this will help a bit in the numbers game. may allow for code injection and other vulnerabilities).

For anonymous server applications, look at what can and can not be done. You can not break a code that does not exist and you can not hide what is in the open. Illusion, obscurity, overload, misdirection are all good ways of handling anonymity. Randomly combining them makes it easy for for only the initiator to know what truly was requested.

Tor, RAID/Torrents, encryption and cypher keys similar to Bible Codes etc

Interesting stuff.

> On February 27th, 2009 phobos said:
>
> You should write up a paper and have it peer reviewed. I think you'll find the current state of research in the areas you've mentioned is further advanced and more complex than your simple overview.

On February 27th, 2009 Anonymous (not verified) said:

I guess many tor nodes are controlled, too many for privacy. You have to do something about it. Inject fake packets or cover traffic, otherwise its broken.
Never mind the net load, we need it in the future, believe me.

> On February 27th, 2009 phobos said:
>
> Your client doesn't trust the nodes already. See, https://blog.torproject.org/blog/circumvention-and-anonymity (https://blog.torproject.org/blog/circumvention-and-anonymity) , for more details.
>
> As for cover traffic, https://wiki.torproject.org/noreply/TheOnionRouter/TorFAQ#YouShouldPad (https://wiki.torproject.org/noreply/TheOnionRouter/TorFAQ#YouShouldPad)
>
> Systems that introduce cover traffic are generally easy to separate the signal from the noise through statistical analysis.

On March 18th, 2009 Anonymous (not verified) said:

While reading about block-level encryption for hard drives, I came across the term "replay attack", although I don't really understand it. I think it means that they repeat the same data more than once.

I remember reading that Tor uses the CBC block cipher mode, all versions of which are vulnerable to that type of attack.

Apparently, the most secure block cipher modes I've read about, CMC and EME, are not

vulnerable to replay-block attacks but are vulnerable to replay-sector attacks. However, since that involved symmetric encryption on a hard disk, I am not sure how well it applies to network encryption.

On March 19th, 2009 Special_Anonymous_someone (not verified) said:

A fancy name and a payload consisting of stuff that is so old everyone has already forgotten it...

On March 21st, 2009 Anonymous (not verified) said:

Good post and good to know the idea from the designer of Tor on a Tor-related paper.

I hope there'll be more similar posts discussing new papers in this field. Thanks.

On April 25th, 2009 Anonymous (not verified) said:

I only use Tor to foil busybody ISPs or government agencies from snooping and logging my surfing habits.

I wouldn't dream of typing identification information (bank accounts, logins, credit card details or even my name) into any forms while using TOR, because of the possibility of compromised exit nodes.

Even using a paid VPN anonymizer, I wouldn't trust them enough to send such information through their servers.

On May 5th, 2009 Malcolm Lambe (not verified) said:

I'm writing some copy for a paid proxy Server and I downloaded and used Tor every day for about a week. UNTIL....my Adwords a/c was hacked and they ran up €400 in clicks before Google and I noticed it - a new account set up with a daily spend of €2000. I suspect they probably got in through my gmail a/c which is obviously linked to the Adwords a/c.

> On May 9th, 2009 phobos said:
>
> Yes, Tor is secure. Google records the last few IP addresses which accessed your account. It's a stretch to say, "I had Tor installed" and blame it for your Google account hack. Did you always login via https? Was your password guessed?
>
> Lots of things could have happened and this could be a coincidence. It's unfortunate that your first experience with Tor coincided with your Google problems.

On July 30th, 2010 Anonymous said:

I can bearly get my 14.4 Zoom modem to connect to compuserve....so all this is a bit techi for me....what I wanna know is will my mother fidn out I am downloading porn? I would hate to go make to wacking off to the underwear section of the old sears catalog.....

On July 31st, 2010 Anonymous said:

It was a very nice idea! Just wanna say thank you for the information you have shared. Just continue writing this kind of post. I will be your loyal reader. Thanks again.
jeux pour gagner des cadeaux
7

On October 5th, 2010 Anonymous said:

Combine two and cache requests for later retrieval of data. End user client will know which data is

theirs, but multiple people will have requested such data on the network at similar times, making it appear more viral than individual. - Issue - others see random real and fictional requests on network and bandwidth increases.
cheap hotels

On May 9th, 2011 Anonymous said:

I guess my question may not get read or answered any time soon but I'm not sure who else to ask so i'll take the chance of wasting a few minutes of my time. I'm doing research to find the most "secure" method of browsing the internet, the goal is to have encrypted, untracable browsing for personal reasons.. My understanding is that this is a pretty tall order, but I'm certain that its not impossible. Thanks for any insight! please forward any responce to my e-mail address burntmacaroni@gmail.com Thanks again!

On September 5th, 2011 Anonymous said:

There is no such thing as untraceable, you just have to minimize the risk of being traced.

On December 21st, 2011 Anonymous said:

As previous commentors have mentioned, use it to hide your IP if you're posting things you wouldn't want the crook class seeing etc, don't use it for private email addresses, credit cards etc.

On February 21st, 2012 Anonymous said:

I so happy very nice your article.i will be your loyal reader.Now you make it easy for me to understand and implement.i love your site Thanks for sharing this.

On May 8th, 2012 Anonymous said:

Are some of these attacks made more difficult if you constantaly click the "Use a New Identity" button," even between every click of a link?

And if you are running via a VPN, changing the country you're coming out of every hour or so, or switching to a different VPN company altogether, back and forth throughout the day?

Post new comment

Comment: *

- Lines and paragraphs break automatically.
- Allowed HTML tags: <cite> <code>   <b> <i> <strike> <p>

More information about formatting options (/filter/tips)

CAPTCHA

This question is for testing whether you are a human visitor and to prevent automated spam submissions.

Math Question: *

3 + 7 =

Solve this simple math problem and enter the result. E.g. for 1+3, enter 4.

Preview comment Post comment