Topic 3: Cloud-based Onion Routing

Advanced Internet Computing, WS2014

Simon Hecht
0926240

Corinna Kirner
0925538

Angela Purker
0926589

Alexander Nebel
0925291

Christian Willinger
0825998

ABSTRACT

Anonymity networks have become hugely popular in the last years. In order to protect ones own privacy such networks are used to communicate within networks anonymously. Due to the goal of protecting people from privacy invasions anonymity networks have powerful enemys. There are many open questions which possible weak points and vulnerabilities do anonymous networks provide for attackers. In order to secure such a system a lot of research has to be done to avoid attacks. Some of them are theoretical others are opened because anonymity networks run in the real world. This paper present a scientific state of the art in anonymity networks. Since the amount of all necessary topics is too huge, we will focus on how to deal with attacks on the privacy and network, defense mechanisms against it and on cloud computing, those topics were relevant parts for our solution of the AIC project [49].

Categories and Subject Descriptors

C.2 [**Computer Communication Networsks**]: Network Protocosl, Internetnetworkting, Miscellaneous

Keywords

Anonymity networks, Tor, Censoreshiped Environments, Fingerprinting, Cloud Computing

1. INTRODUCTION

Anonymity systems can be divided into high and low latency systems. High latency systems uses long delivery time which reaches from hours to days. Examples for high latency systems are remailers [27, 51, 32, 13] which strips informations about users and than send mails to the target. An important notable development in these systems were made in 1981 in which David Chaum introdruced Chaum mixes [9] which changes the order of messages and adds padding. That was made possible by asymmetric encryption. High latency systems are not useable for protocols which needs a fast response like HTTP, SSH, etc. They use time for making network analysis more difficult which low latency systems can't. A lot of topics described in later sections deals with that subject. Single systems like proxies, bouncer, VPN services are easy to monitor so the idea of David Chaum has been taken up too and is used in Onion Routing like in the Tor network [15].

Onion routing takes a route through several nodes of the anonymity network and adds an encryption layer (onion layer) to each node so that through the path each node can only decrypt its layer with the routing information. The last node sends the unencrypted package into the network. Garlic routing developed by Michael J. Freedmanin et al. [16] collects messages to garlic cloves and sends them together as used in the I2P project [52]. A general problem of all these networks is that an global observer can analyse traffic, there are systems based on the dining cryptographers problem [14] like the DC Net [8] and broadcast systems like P5 [11]. P2P anonymity systems like Freenet [46] or GNUNet [20, 21] tries to be censor resistent but they still lack on scalability on routing [10].

We have focused this paper on topics which could be relevant for our project if we would make it global [Note: we will never do that]. In section 2 we will discuss some current topics and approaches on anonymity networks. How to deal with censorshiped environments is discussed in section 2.1. In section 2.2 we describe node reputation and browser fingerprinting. Section 2.3 deals with fingerprinting in general. In section 2.4. we discuss how botnets can abuse the network and what can be done against it, and section 3 concludes the paper.

2. TOPICS ON ANONYMITY NETWORKS

2.1 Communicate within censorshiped environements

For an anonymity network like *Tor* it can be a problem to communicate with the anonymity network. Countries who censor their network can try to identify entry points into the anonymity network, block connections to that networks and identify their users. There are several approaches to deal with that problem where we will discuss the most efficient ones in the following subsections. Furthermore a broad overview can be found in the *Tor* project's svn [38].

2.1.1 Decoy routing

Decoys [29, 25, 57] are the idea to use allied routers (decoy routers) in the internet infrastructure to redirect traffic which is part of an anonymity network to the proxy(decoy proxy) target while it looks like it's normal traffic to another destination. A client can signal decoy routers to redirect it's TCP connection. Karlin et al.

suggested that this can be done by port knocking [29], symmetric encrypted data in the payload or a series of payload lenght but they also introduced a technique to hide it in a TLS handshake. When decoy routing is detected which is possible through inconsistence path behaviour the client can be blocked and decoy routers can also be blocked when identified by a routing around decoys attack(RAD) [48]. Houmansadr et al. [26] simulated base on real internet data that the network costs for a country can be very high when decoy routers are well placed on their network infrastracture. Their results showed that stratetical played decoy routers on 1% of the autonomous systems can disconnect China from 18%, Venezuela from 54% and Syria from 87% of all internet destinations.

2.1.2 Cloud computing

Brubaker et al. [4] introduced the idea that to use cloud storages services such as Amazon S3 to provide entry points into anonymity networks. They built a system named CloudTransport which can be used as a standalone application, a gateway to an anonymity network or as an pluggable transport for *Tor*. The autors argue, that countries like China could be unwilling [22] to shut down cloud services because of the risk of shutting down other encrypted services which are important and willed by people in the country. To do so they have used the same libaries and have connected encrypted to the cloud as other cloud services do.

2.1.3 Tor bridges

Networks like *Tor* contains public trusted directory authorities which provides *Tor* nodes to use the network. A list of this nodes can be easily generated by an attacker like China [30] who wants to block this clients. Because of that the *Tor* project invented so created bridges [41] as entry nodes which are not publicly known. They can be requested by email or by solving a captcha. [42] zmap is a tool by Durumeric et al [17] which allows to theoreticaly scan the IPv4 range within 45 minutes with a gigabit internet connection. With these tool they could identify 87% of the bridges in the internet by scanning for *Tor* handshakes on port 9001 and 443. A countermeassure for this is random ports selection on bridge announcement as obfsproxy which is explained later in this section. China is reportedly trying to locate *Tor* bridges [43, 55].

Bridge operators can be identified connection to a website several times [31] by a circuit-clogging attack [34].

To make it more difficult to identify bridges in the *Tor* network Smits et al. [50] uses a single packet for authorization which makes connections from attackers more difficult. To make connection detection to bridges more difficult Appelbaum and Mathewson [1] created a framework for pluggable *Tor* transports. With this it's possible to make the connection looks like something else, Mathewson's obfsproxy [44]. âĂŽ uses an encrypted stream for that. Moghaddam et al [33] built an obfuscation system which makes connections to the *Tor* network look like Skype video traffic. David Fifield [45] wrote a pluggable transport called meek which uses the technique domain fronting which makes traffic appear like it comes from a big domain like google or amazon.

2.1.4 P2P

An approach would be the usage of Freenet. a peer-to-peer platfrom and decentralized publication system designed for censorhip-resistant communication. By measuring the censorship-resilence of Freenet despite its obfuscation protocols Roos et al. [47] proves the existence of bottlenecks of the existing algorithms used by Freenet. The result shows that the current topology control mechanism is suboptimal and insufficient for routing due to their neighbor selection and its interaction between Opennet and Darknet.

The most striking difference to other peer-to-peer systems is that Freenet is used by a huge amount of users who exhibit uncharacteristically long online times, a session length of more than 90 minutes.

2.2 Node reputation and browser fingerprinting

Bad quality of nodes in an anonymity network can degrade the user experiance or even be malicious for an user. Das et al. [12] created a reputation system for anonymity system which rates nodes who do time variation for attacks down over time and developed a filtering strategy. Traffic in anonymity networks like onion networks can be observed and manipulated by the exit node which makes the connection to the target by design. Non encrypted connections can be watched, MIM-attacks like sslstrip can be done on TLS. Also attacks on a web user like browser fingerprinting [37]can be done. Because of all that the *Tor* project created the *Tor* browser bundle which should creates a browser environment which protects the user from attacks on his privacy. It uses the extension NoScripts which should prevent script attacks and HTTPS-Everywhere which changes HTTP-connections to HTTPS connections. However there a lot of attacks from which the *Tor* browser can't protect. Attacks like on weak chiper suits, session cookies, heartbleed [23], non-HTTPS traffic like SSH are still possible. Winter et created methods to scan and identify malicous *Tor* exit nodes. They built HoneyConnector, a framework to detect sniffing *Tor* exit node and exitmap a fast exit node scanner. Winter et al. also patched the torbutton so that it compares X.509 certificates through different *Tor* routes. With that methods they identified 65 exit relays which can be marked as BadExit [56].

2.3 Fingerprinting

Website fingerprinting is described by Wang et al. [53]. A passive eavesdropper can monitoring size lenght or timing information and fingerprint which website a user is using. Their team trained a software to monitor 100 web pages by classifing their patterns and recognised the usage in an simulation with also websites which are not trained. Attacks on *Tor* are more difficult than on other networks because the project uses cell padding and background noise. They also simulated the optimal defense with a good bandwith overhead relation. Related work on website fingerprinting described by them were Resource length attacks which worked on HTTP 1.0 because the protocol isn't using persistant connection as HTTP 1.1 does. Also unique packet length attacks can also be used to fingerprint websites by lenght. In 2009 Panchenko et al. [36] used hidden packet length attacks on *Tor* to identify unique packet lenghts which are hidden by the *Tor*'s cell padding. They were been further improved [18, 5, 54] Murdoch and Danezis [34] developed a practical traffic analysis in 2005 on *Tor*. Evans et al. stated later that this technique isn't accurate anymore because of the increased traffic [19]. In 2007 Murdoch [35] used NetFlow data to analyse traffic from the *Tor* network. On 2014 Chakravart et al. [7] used a NetFlow data analyse technique where they injected a traffic pattern in the server's response and computed the correlation. They claimed to have 100% accurancy in their lab environment and 81.6% accurancy in the tor network with a value of 5.1% false positive.

However the *Tor* project shows in their blog entries that these results shouldn't be overrated because of the false positives in the results [2, 3].

2.4 Botnet

At the end of 2013 the mevade botnet begun to use it's C & C servers over tor to anonymise their communication. While the traffic hadn't increased much it put a lot of load on tor relays to built circuits. They create so called hidden services [40] which allows to host services anonymously in the network [39]. The *Tor* project reacted to this by releasing ntor to exchange keys more efficient. Nicholas Hopper analysed it in 2014 [24] long term strategies against botnet abuse. He suggested to add costs for building circutes for hidden services like human attention(CAPTCHAs), processor time, bitcoins, etc and to ensure that this can't be double spent. There are serveral problems he found out about this approaches, as it's a anonymous network and without blacklisting, bot's can just requery new challanges to solve. Also bots can do "chain-proving" as chain voting does [28] to solve challenges. He also suggested to limit the rate of the entry guard which's disadvantage is that it could affect other hidden services and he also brought the idea to isolate hidden service circuits from other circuits. Other research goes in the of de-anonymization botnets. Casenove and Miraglia published Botnet over Tor: The Illusion of Hiding [6] in 2014 where they claimed that *Tor* doesn't bring the necessary anonymity for a botnet.

3. CONCLUSIONS

A lot of systems and strategies occurs which tries to identify a user in anonymity networks, but there are a lot of countermeasures to protect the users identity. For eyample there are many good and practicable solutions when dealing with the problem of communication within censorshiped environements and should be kept in mind during the implementation of an anonymity network. Also the quality of the nodes can be checked by using a node reputation system for detecting and identifying malicoius Tor exit nodes in order to avoid them. Anonymity networks will become an even more popular research area in the future.

4. REFERENCES

[1] Jacob Appelbaum and Nick Mathewson. Pluggable transports for circumvention, 2010.

[2] Tor Project Blog. Tor Project - Critiwue website traffic fingerprinting attacks. `https://blog.torproject.org/blog/critique-website-traffic-fingerprinting-attacks`.

[3] Tor Project Blog. Tor Project - Traffic correlation using netflows. `https://blog.torproject.org/blog/traffic-correlation-using-netflows`.

[4] Chad Brubaker, Amir Houmansadr, and Vitaly Shmatikov. Cloudtransport: Using cloud storage for censorship-resistant networking.

[5] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM conference on Computer and Communications Security*, pages 605–616. ACM, 2012.

[6] Matteo Casenove and Armando Miraglia. Botnet over tor: The illusion of hiding. In *Cyber Conflict (CyCon 2014), 2014 6th International Conference On*, pages 273–282. IEEE, 2014.

[7] Sambuddho Chakravarty, Marco V Barbera, Georgios Portokalidis, Michalis Polychronakis, and Angelos D Keromytis. On the effectiveness of traffic analysis against anonymity networks using flow records. In *Passive and Active Measurement*, pages 247–257. Springer, 2014.

[8] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of cryptology*, 1(1):65–75, 1988.

[9] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[10] Ian Clarke, Oskar Sandberg, Matthew Toseland, and Vilhelm Verendel. Private communication through a network of trusted connections: The dark freenet. *Network*, 2010.

[11] cs umd edu. Broadcast systems P5. `http://www.cs.umd.edu/projects/p5/`. Accessed: 2015-01-25.

[12] Anupam Das, Nikita Borisov, Prateek Mittal, and Matthew Caesar. Re 3: relay reliability reputation for anonymity systems. In *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pages 63–74. ACM, 2014.

[13] Mixminion Minion Design. High latency system example. `http://www.mixminion.net/minion-design.pdf/`. Accessed: 2015-01-25.

[14] EW Dijkstra. The ew dijkstra archive. Technical report, Technical report, http://www. cs. utexas. edu/users/EWD, 2009.

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

[16] Roger R Dingledine. *The free haven project: Design and deployment of an anonymous secure data haven*. PhD thesis, Citeseer, 2000.

[17] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. Zmap: Fast internet-wide scanning and its security applications. In *USENIX Security*, pages 605–620. Citeseer, 2013.

[18] Kevin P Dyer, Scott E Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 332–346. IEEE, 2012.

[19] Nathan S Evans, Roger Dingledine, and Christian Grothoff. A practical congestion attack on tor using long paths. In *USENIX Security Symposium*, pages 33–50, 2009.

[20] Gnu. Gnunet. `https://gnunet.org/`. Accessed: 2015-01-25.

[21] Gnu. Gnunet. `https://gnunet.org/whitepaper`. Accessed: 2015-01-25.

[22] The guardian. The guradian Article. `http://www.theguardian.com/world/2013/nov/18/`. Accessed: 2015-01-25.

[23] Heartbleed. Heartbleed. `http://heartbleed.com/`.

[24] Nicholas Hopper. Challenges in protecting tor hidden services from botnet abuse.

[25] Amir Houmansadr, Giang TK Nguyen, Matthew Caesar, and Nikita Borisov. Cirripede: circumvention infrastructure using router redirection with plausible deniability. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 187–200. ACM, 2011.

[26] Amir Houmansadr, Edmund L Wong, and Vitaly Shmatikov. No direction home: The true cost of routing around decoys. 2014.

[27] Informatik Huberlin. High latency system example.

http://waste.informatik.huberlin.de/Grassmuck/Texts/remailer.html. Accessed: 2015-01-25.

[28] Douglas W Jones. Chain voting. In *Workshop on Developing an Analysis of Threats to Voting Systems, National Institute of Standards and Technology*, 2005.

[29] Josh Karlin, Daniel Ellard, Alden W Jackson, Christine E Jones, Greg Lauer, David P Mankins, and W Timothy Strayer. Decoy routing: Toward unblockable internet communication. In *USENIX Workshop on Free and Open Communications on the Internet*, 2011.

[30] Andrew Lewman. China blocking tor: Round two, 2010.

[31] Jon McLachlan and Nicholas Hopper. On the risks of serving whenever you surf: vulnerabilities in tor's blocking resistance design. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 31–40. ACM, 2009.

[32] Mixminion. High latency system example. http://www.mixminion.net/. Accessed: 2015-01-25.

[33] Hooman Mohajeri Moghaddam, Baiyu Li, Mohammad Derakhshani, and Ian Goldberg. Skypemorph: Protocol obfuscation for tor bridges. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 97–108. ACM, 2012.

[34] Steven J Murdoch and George Danezis. Low-cost traffic analysis of tor. In *Security and Privacy, 2005 IEEE Symposium on*, pages 183–195. IEEE, 2005.

[35] Steven J Murdoch and Piotr Zieliński. Sampled traffic analysis by internet-exchange-level adversaries. In *Privacy Enhancing Technologies*, pages 167–183. Springer, 2007.

[36] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114. ACM, 2011.

[37] Panopticlick. Fingerprinting. https://panopticlick.eff.org/. Accessed: 2015-01-25.

[38] Tor Project. Design paper blocking. https://svn.torproject.org/svn/projects/design-paper/blocking.html. Accessed: 2015-01-25.

[39] Tor Project. How to hande millions new tor clients. https://blog.torproject.org/blog/how-to-handle-millions-new-tor-clients. Accessed: 2015-01-25.

[40] Tor Project. Project Tor Design. https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf. Accessed: 2015-01-25.

[41] Tor project. Tor project - Bridges. http://gitweb.torproject.org/torspec.git/tree/dir-spec.txt?id=3c0422acc464a9da74bc35d3027ae966bf59d8d0. Accessed: 2015-01-25.

[42] Tor project. Tor project - Bridges. https://www.torproject.org/docs/bridges.html.en. Accessed: 2015-01-25.

[43] Tor project. Tor project - Knock Knock Knockin Bridges doors. https://www.torproject.org/docs/bridges.html.en. Accessed: 2015-01-25.

[44] Tor project. Tor project - Obfuscation Proxy. https://gitweb.torproject.org/obfsproxy.git. Accessed: 2015-01-25.

[45] Tor project. Tor project - Pluggable Transport. https://trac.torproject.org/projects/tor/wiki/doc/meek. Accessed: 2015-01-25.

[46] Freenet projects. Freenet Project. https://freenetproject.org/papers/freenet-0.7.5-paper.pdf. Accessed: 2015-01-25.

[47] Stefanie Roos, Benjamin Schiller, Stefan Hacker, and Thorsten Strufe. Measuring freenet in the wild: Censorship-resilience under observation. In *Privacy Enhancing Technologies*, pages 263–282. Springer, 2014.

[48] Max Schuchard, John Geddes, Christopher Thompson, and Nicholas Hopper. Routing around decoys. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 85–96. ACM, 2012.

[49] Angela Purker Alexander Nebel Christian Willinger Simon Hecht, Corinna Kirner. Cloud Based Onion Routing AIC WS2014 Project. https://github.com/n-n-nebbl/Cloud-Based-Onion-Routing. Accessed: 2015-01-25.

[50] Rob Smits, Divam Jain, Sarah Pidcock, Ian Goldberg, and Urs Hengartner. Bridgespa: improving tor bridges with single packet authorization. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 93–102. ACM, 2011.

[51] Sourceforge. High latency system example. http://mixmaster.sourceforge.net/. Accessed: 2015-01-25.

[52] Mater thesis davor. High latency system example. https://geti2p.net/de/. Accessed: 2015-01-25.

[53] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting.

[54] Tao Wang and Ian Goldberg. Improved website fingerprinting on tor. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 201–212. ACM, 2013.

[55] Philipp Winter and Stefan Lindskog. How china is blocking tor. *arXiv preprint arXiv:1204.0447*, 2012.

[56] Philipp Winter and Stefan Lindskog. Spoiled onions: Exposing malicious tor exit relays. *arXiv preprint arXiv:1401.4917*, 2014.

[57] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J Alex Halderman. Telex: Anticensorship in the network infrastructure. In *USENIX Security Symposium*, 2011.