A look into Security Issues in P2P Networks

University of Wisconsin-Madison

Thomas Hansen, Akash Khetwani
(May 2017)

Table of Contents

P2P Networks

The internet has changed significantly over the years. It has gone from a small network of experimental computers to a massive ecosystem of nodes communicating together. It would only make sense that some of these networks would be peer-to-peer, a paradigm that now makes up 15-20% of all web traffic. [1]

Peer-to-Peer (P2P) networks are defined as those who exhibit three characteristics: self-organization, symmetric communications and distributed control. [2] They are popular as they allow for networks to quickly scale, running partly or completely from the users contributing content and/or bandwidth. P2P networks are able to build a resource-rich system by aggregating the resources of a large number of independent nodes that enables these systems to dwarf the capabilities of many centralized system for little cost. Many are centralized at some point (such as bittorrents trackers, which maintain lists of clients and downloadable files) however the majority of the work is done between client nodes. [3]

Traditionally P2P networks have been typically used for file sharing applications, which enable peers to share digitized content such as general documents, audio, video, electronic books, etc. Recently, more advanced applications such as real-time conferences, online gaming, and media streaming have also been deployed over such networks.

There are usually four different areas people research in P2P literature: search, security, storage, and applications. [4] We will primarily be looking into security and attacks on P2P Networks, which can be a major issue due to the decentralized nature of P2P Networks.

P2P Background

P2P Networks are used in many different applications, with different amounts of integration into the application, from just messaging to transferring large files.

These networks do cause excess strain on ISP networks, primarily due to a higher amount of upstream bandwidth being used. Due to them increasing network strain and often transporting illegal content, ISP's are incentivised to block this content. [5] This generates one of the largest sources of attacks on these networks, and when combined with other issues this means P2P networks need to be secure, and are often under or being attacked.

Attacks on P2P Networks

Since P2P systems inherently rely on the dependence of peers with each other, security implications arise from abusing the trust between peers. In a traditional client-server model, internal data need not be exposed to the client, but with P2P, some internals must be exposed

to fellow peers in the name of distributing the workload. Attackers can leverage this in compromising P2P networks.

Packet dropping

One way ISP's fight P2P traffic is by dropping packets sent through ports commonly used in P2P applications using deep packet inspection (DPI). Although DPI can be done in a number of ways, it is most often done using string matching sometimes through physical hardware. One method, called bloom filters, are even able to do this without decreasing network throughput and without ever generating false negatives. This technology is not specific to P2P traffic, and is still being researched and worked on to prevent malicious packets from damaging the network. [6]

P2P clients also fight ISPs by sending data through different ports, as many modern P2P applications can send data through port 80 or even choose them randomly. [7] Additionally clients often encrypt or obfuscate both the data and the header, so that the ISP is unable to read what kind of data is being sent. [5] What's notable here is that the encryption isn't intended to prevent anyone from reading the content, just strong enough to avoid the payload string matching done by ISP's to target P2P traffic from clients such as bittorrent.

There are more utilitarian options that have been researched for preventing dropping packets as well. Some research done by Choffnes and Bustamante involves reducing how much outgoing traffic that's caused by the P2P client. They posit that by tracking which CDN's each user uses, they can guess the relative location of the peer, and exchange data to those who are closer avoiding bottlenecks that occur during cross ISP traffic and lessening the load on ISP's. In their proof of concept plugin Ono, it increased the likelihood traffic stayed in one automated system from ~10% to 33%. This takes a different approach to packet dropping, making their system more friendly to ISPs and reducing the likelihood they'll continue to try and stop P2P traffic. [7]

Packet dropping as a field however is not being researched as much anymore, due to the superior ability of P2P clients to hide P2P traffic, and most research has stopped since 2010.

DoS/DDoS

There are two main different ways people can try to take down these P2P networks; DoS and DDoS. In a DoS (Denial of Service) attack, a single host tries to flood the network with requests so that it becomes to congested to function and possibly crashes. In a DDoS, many different nodes start making requests in an attempt to damage certain nodes or even take down the entire network provided it's small enough

Currently the most commonly researched version is DDoS, however much of the research is from around 2010 and DoS are usually more common. One example of a DoS attack on a Gnutella network involves a malicious node getting a central position within the network. Once it's there, it's able to recommend all queries to a certain node, even though the node doesn't

have the files requested in the query. Attacks like these can be limited, however require more rigorously constrained protocols. [8]

Attacks like these are more easily executed on unstructured networks, one of two major paradigms in P2P architecture. The other of course being structured. Unstructured networks are especially hard to find attackers and even harder to stop attackers. By attacking an unstructured network you leave other items such as HTTP exploitation and query flooding as easy options for attackers to abuse and damage the network. [9]

One of the most effective ways to try to prevent these kinds of attacks is by using firewalls/IP filtering. Since you have a large number of nodes, expensive functions may be farmed out to keep the firewall up while additionally not revealing which nodes are hosting/doing a lot of the processing. These filters however do become a target, and some systems such as SOS (secure overlay services) recommend hosting them inside ISP's, since their routers could more easily handle the large amount of traffic they may receive. It also provides a method of clear communication between the target node and a confirmed user. [10] A wide array of options exist, however many of these solutions are in production and no longer being extensively researched.

In some cases traffic from P2P networks can even be used to cause DDoS attacks against other nodes/servers. Due primarily to the spread of clients and potential number of TCP packets, one study found you could use set victim servers as bittorrent trackers (clients that re-route traffic) and use the proceeding flood of traffic to slow/take down the victim computer/server. [11]

File Authenticity

File authenticity is different than many other forms of attack, in that it's compromising the integrity of the information, thus leaving the network intact but the contents worthless or dangerous for the user. The most common way this is tackled is by adding a key to each file, or another level of redundancy. Some methods that include using a key are CRCs (cyclic redundancy checks), Hashing, MACs (message authentication codes), and digital signatures, however other, non-key-based methods are used. [12] The theme among these options is that you can decrypt and verify the key is still correct, thus suggesting the file is still correct as well.

Other tricks such as oldest-document may be used, involving verifying the metadata with the new document to ensure it's the same. Reputation and voting-based systems both rely on deciding some nodes to be more valuable than others. Expert-based systems usually rely on a governing-system to decide who is and is not trustworthy. [13] Lastly it is possible to keep a central login server which significantly reduces the network costs and saves time, however this does force us to maintain a trusted server. [14] We also see a mixture of these as well. Services such as the blockchain in bitcoin network's first use oldest-document and then verify

transactions using a voting-method to decide which transactions to onto the blockchain. The nodes who've downloaded the blockchain now act as an expert when determining which transactions actually happened and which were forged.

File authenticity was largely based off of existing technology, thus a lot of the research in items such as CRC and hashing has long since matured, and research specifically in P2P file authenticity has mostly died down. This is still an issue that could use better solutions however as many of the options listed above have issues, such as requiring some nodes to be more important than others or are still manipulatable by ill intentioned nodes in the network.

Securing a P2P network

P2P systems lack the tools available to a centralized administrator as in a client-server model. Thus, it can be much more difficult to implement security protections on a deployed P2P system. Moreover, the absence of a defensible border of a system means that it is hard to know friend from foe. We have discussed the security threats related to P2P networks. Now, we look at methods to securing these networks.

Encryption

Encryption is the process of creating a secure communication link between the two peers. This protects the sensitive information transiting the network from intermediate devices. By encrypting P2P traffic, the hope is that not only will the data be safely encrypted, but more importantly, the P2P data stream is encrypted and not easily detectable.

Historically there have been issues keeping private keys secret across P2P Networks since many clients will have to be able to use/view the keys. Encrypting files before swapping them isn't a reliable method to mask online activity, according to Internet security experts. Decryption keys are readily available, especially to experts whose jobs involve intercepting encrypted data that is part of file-swapping activities. Experts say it is quite common for investigators to trap encrypted files from peer-to-peer networks and determine the content. It's in large part for this reason that few networks have strong encryption standards, and those that do are often related more towards messaging then file sharing. This said in many cases the encryption only has to act as obfuscation so the ISP can no longer match strings with banned strings in the header, which could technically be done by scrambling any headers together. [5]

Encryption is becoming more popular in messaging and media P2P networks, but with the help of some servers. Services such as skype use a strong 256 bit encryption, however are aided by central servers that help begin the connection. [16] Areas such as these are newer, however are being build off of previous encryption knowledge and are not the cause of much research.

Encryption is important for two reasons: it protects the data from people trying to read what's being sent, and it protects against ISP's dropping packets, usually using RIAA as an excuse to disallow bandwidth-heavy P2P traffic. [1] With the actual connection stream completely encrypted, it becomes much harder for the P2P traffic to be detected, and, thus, attacked, blocked, or throttled. A very good example of development in this arena is encrypted BitTorrent, which can encrypt both the header and the payload. Using only 60-80 bits for the cipher, the aim is not to protect the data but instead to simply obfuscate the stream enough so that it is not detectable without incurring much of a performance hit. Although it is still possible to detect encrypted BitTorrent streams using sophisticated methods based on pattern and timing of the

traffic, in practice, it is much harder to filter encrypted streams now.Encryption of P2P traffic seems to be picking up, as currently about 20% of BitTorrent traffic is encrypted. [17]

Anonymity

An anonymous P2P communication system is a peer-to-peer distributed application in which the nodes or participants are anonymous or pseudonymous. Anonymity of participants is usually achieved by special routing overlay networks that hide the physical location of each node from other participants. [18]

The holy grail of anonymity is to not use the user's IP address, however from a practical standpoint this is impossible, as it has to be traceable from some point. Most work and research in the area is done towards obfuscation, or making it difficult to track and catch P2P packets. [19] In terms of masking the user's IP address in P2P networks, this can be tricky due to the decentralized nature of the network. Stegtorrent encodes the packets and then has them sent in a non uniform order to a network of nodes who then discretely forward it to the requesting node. Although this of course introduces a small delay, it does make it significantly harder to track and find torrent packets, protecting the people using the network. [20]

Each node selects from what it can see of the network, a set of peers to act as mimics. Initial node discovery and subsequent network maintenance is based on a gossip model, where nodes share their view of the network. Mimics are selected randomly from the available nodes for security and balance, in some verifiable manner. Each node exchanges a constant rate of cover traffic of fixed size packets with its mimics using symmetric encryption. Symmetric keys are distributed using the relays' public keys. Actual data can now be interwoven into the cover traffic without an observer detecting where a message originates. [21]

To build a path, the sending node randomly selects a given number of mimics and wraps the message in an "onion" of symmetric keys from each node on the path. The sender passes the packet—outwardly indistinguishable from cover traffic—to the first node in the chain, which removes the outermost wrapper with its private key, and then sends it along to the next node. With the exception of the last node, each node in the chain is aware of the node before and after it in the chain, but has no way of telling where it is in the chain itself. That is, the node cannot tell if it is the first hop or the penultimate hop. The final node in the chain of mimics acts as the Network Address Translator for the transport layer, and sends the packet to its final destination through the Internet. This final node must know the content and destination, but has no information about the sender.

Nodes store a record for the return path, so a reply from the Web host contacted can be received by the final node in the chain, rewrapped with its private key, and sent back to the penultimate hop. The message is then passed back through the chain, with each node adding another layer of encryption. The originating node can use the public keys of each node to

unwrap the layers and read the message. Since it is the only node to know the public keys of each hop along the path, the content is secure. [22]

Due to many legal/tracking concerns, anonymity can be very important in P2P systems, although it creates some interesting problems. Specifically introducing anonymity makes it difficult to track reputation and prevent harmful clients from easily clearing their reputation to start again. [23]

Summary

Security in P2P networks has been a major and significant issue, however it's concern recently has been minimized. These networks are discriminated against by ISP's and other content providers in ways other networks don't experience, and have in turn been forced to create new and creative ways to avoid detection and being damaged or taken down. Over time, however, these methods have become standard as the field has matured and the amount of new research coming out on either end has begun to slow down.

Citiations

[1] T. Karagiannis, A. Broido, M. Faloutsos, K. Claffy. “Transport Layer Identification of P2P Traffic”. 2004. Available: http://dl.acm.org/citation.cfm?id=1028804
[2] N. Daswani, H. Garcia-Molina, and B. Yang. “Open Problems in Data-Sharing Peer-to-Peer Systems”. International Conference on Database Theory. December 16, 2002. Available: https://link.springer.com/chapter/10.1007/3-540-36285-1_1
[3] M. Sirivianos, J. Park, H. Chen, X. Yang. (2007, February). “Free-riding in BitTorrent Networks with the Large View Exploit.” In IPTPS. Retrieved from http://ai2-s2-pdfs.s3.amazonaws.com/77d2/2d1c4ed79751e384b0d1ff75497a7a885afc.pdf
[4] J. Risson, T. Moors. "Survey of Research Towards Robust Peer-to-Peer Networks: Search Methods." Technical Report, University of New South Wales, Sydney, Australia. 2004. http://www.cs.umd.edu/projects/p2prg/p2p-overview.pdf
[5] T. Karagiannis, A. Broido, N. Brownlee, M. Faloutsos, and K. Claffy. “Is P2P dying or just hiding? [P2P traffic measurement]”. 2004. Available: http://ieeexplore.ieee.org/abstract/document/1378239/
[6] Dharmapurikar, S., Krishnamurthy, P., Sproull, T., & Lockwood, J. (2003, August). Deep packet inspection using parallel bloom filters. In High performance interconnects, 2003. proceedings. 11th symposium on (pp. 44-51). IEEE. Retrieved from http://ieeexplore.ieee.org/document/1231477
[7] D. R. Choffnes, F. E. Bustamante. "Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems". ACM SIGCOMM Computer Communication Review, vol. 38, no. 4, pp. 363-374. ACM, 2008.
[8] N. Daswani, H. Garcia-Molina, and B. Yang. “Open Problems in Data-Sharing Peer-to-Peer Systems”. International Conference on Database Theory. December 16, 2002. pp. 9. Available: https://link.springer.com/chapter/10.1007/3-540-36285-1_1
[9] E. Athanasopoulos, K. Anagnostakis, E. P. Markatos. “Misusing Unstructured P2P Systems to Perform DoS Attacks: The Network That Never Forgets”. International Conference on Applied Cryptography and Network Security. 2006. Available: https://www.ics.forth.gr/_publications/gdos.acns06.pdf
[10] A.D. Keromytis, V. Mirsa, D. Rubenstein. “SOS: An architecture for mitigating DoS attacks”. IEEE. Volume 22, Issue 1, Jan 2004. Retrieved from: http://ieeexplore.ieee.org/document/1258124
[11] K. Defrawy, M. Gjoka, and A. Markopoulou. “BotTorrent: Misusing BitTorrent to Launch DDoS Attacks”. 2007. Available: https://www.usenix.org/legacy/event/sruti07/tech/full_papers/eldefrawy/eldefrawy.pdf
[12] N. Daswani, H. Garcia-Molina, and B. Yang. “Open Problems in Data-Sharing Peer-to-Peer Systems”. International Conference on Database Theory. December 16, 2002. pp. 10. Available: https://link.springer.com/chapter/10.1007/3-540-36285-1_1

[13] N. Daswani, H. Garcia-Molina, and B. Yang. “Open Problems in Data-Sharing Peer-to-Peer Systems”. International Conference on Database Theory. December 16, 2002. pp. 10-11. Available: https://link.springer.com/chapter/10.1007/3-540-36285-1_1
[14] S. Marti, H. Gracia-Molina. “Identity crisis: anonymity vs reputation in P2P systems”. Proceedings Third International Conference on Peer-to-Peer Computing. 2003. Available: http://ieeexplore.ieee.org/document/1231513
[15] S. Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. Available: https://bitcoin.org/bitcoin.pdf
[16] G. Lisha, L. Junzhou, “Performance Analysis of a P2P-Based VoIP Software”, IEEE, Feb. 2006.
[17] Moors Tim, Risson John. “Survey of Research towards Robust Peer-to-Peer Networks: Search Methods”. Available: http://www.cs.umd.edu/projects/p2prg/p2p-overview.pdf
[18] T. Zink, M. Waldvogel. (2012, September). “BitTorrent traffic obfuscation: A chase towards semantic traffic identification.” In Peer-to-Peer Computing (P2P), 2012 IEEE 12th International Conference on (pp. 126-137). IEEE.
[19] B.Suvarna Raju, B.Durga Sri. “A Novel Data Distribution Scheme for VoD Services in P2P Networks”. (September 2013). Available: www.ijrcct.org/index.php/ojs/article/download/368/284
[20] Freedman Allan, L Jean Camp. “Peer-to-Peer Security”. Available: http://allan.friedmans.org/papers/P2Psecurity.pdf
[21] P. Kopiczko, W. Mazurczyk, K. Szczypiorski. (2013, May). “Stegtorrent: a steganographic method for the p2p file sharing service.” In Security and Privacy Workshops (SPW), 2013 IEEE (pp. 151-157). IEEE.
[22] Freedman M.J., & Morris R. Tarzan. “A Peer-to-Peer Anonymizing Network Layer”. ACM Conference on Computer and Communications Security (CCS 9). Washington, D.C.
[23] R. Dingledine, N. Mathewson, P. Syverson. “Reputation in P2P Anonymity Systems”. 2003. Available: https://www.freehaven.net/anonbib/cache/rep-anon.pdf

Acronyms/Initializations used:

DDoS: Distributed Denial-of-Service
DoS: Denial of Service
ISP: Internet Service Provider
P2P: Peer-to-Peer
RIAA: Record Industry Association of America
TCP: Transmission Control Protocol