Peer-To-Peer Security

MOHD ASRI BIN MOHAMAD STAMBUL
UNIVERSITI KEBANGSAAN MALAYSIA
Bangi , Selangor
Email: mdasri.stambul@yahoo.com

Abstract

Peer-to-peer (p2p) networking technologies have gained popularity as a mechanism for users to share files without the need for centralized servers. A p2p network provides a scalable and fault-tolerant mechanism to locate nodes anywhere on a network without maintaining a large amount of routing state. This allows for a variety of applications beyond simple file sharing. Examples include multicast systems, anonymous communications systems, and web caches. the review are about the security issues that occur in p2p applications.

1. Introduction

Peer-to-peer systems, beginning with Napster, Gnutella, and several other related systems,became immensely popular in the past few years, primarily because they offered a way for people to get music without paying for it. However, under the hood, these systems represent a paradigm shift from the usual web client/server model, where there are no servers; every system acts as a peer, and by virtue of the huge number of peers, objects can be widely replicated, providing the opportunity for high availability and scalability, despite the lack of centralized infrastructure. Peer-to-peer systems (P2P) have grown in importance over the last 5 years as an attractive way to mobilize the resources of Internet users. As more and more users have powerful processors, large storage spaces and fast network connections, more actors seek to coordinate these resources for common goals. Because of their unique decentralized nature, security in these systems is both critical and an interesting problem. Good security on P2P systems must reflect the design goals of the system itself. The introduction of some basic concepts for P2P systems and their security demands, and then discuss several case studies to highlight issues of robustness, privacy, and trust. An analysis of the systems that offer secure distributed routing, privacy-enhancing and censorship-resistance publishing, shared storage and decentralized collaborative work spaces. The conclusions are about an accountability and trust mechanisms must be built into a secure system.

2. Peer-To-Peer Systems

Peer-to-peer systems have two dominant features that distinguish them from a more standard client-server model of information distribution: they are overlay networks that have unique namespaces. P2P systems link different, possibly heterogeneous systems as peers, and allow them to interact on top of existing network configurations. It does this by defining relationships unique to that system, usually in the form of a topology by which systems are linked. Occasionally a system will piggyback atop an existing namespace, such as the IP/port labeling, but still treats these separately from the Internet-layer protocols. The combined effect of independent systems interacting through a unique namespace is decentralization. Not every system generally considered a P2P system is strictly decentralized in managementhe infamous Napster actually ran through a centralized serverbut the matter of decentralization brings out important concerns for system security. Although they have won the most attention for their roles as (often illicit) file-swapping networks, peer-to-peer systems can serve many functions, and design considerations must reflect these. One proposed use of a P2P system was that of shared expertise. A Gnutella pioneer has suggested that the system could be used to propagate queries of any sort through a network until a peer felt that it could respond to the query, anything from a lexical look-up to a specialized calculation (Kan, 2000). In a network with varied resource availability, P2P systems have been used to distribute those resources to those who need or want them the most. The shared resources in question are usually computation or storage. This can create efficient resource usage, given a low marginal cost of using a resource not otherwise needed. Beyond sharing computation power for enormous tasks, P2P networks have been proposed as an escrow system for digital rights management, or for distributing searches across a wide range of other peers. A new class of business software systems known as groupware uses many P2P principles. Groupware networks are closed networks that support collaborative work, such as the Groove network discussed below. Finally, the decentralized nature of peer systems offers many positive features for publishing and distribution systems, not least of which is their resilience to legal and physical attacks and censorship. P2P architectures have many different functions, and different functions lead to

different design conclusions, with respect to overall system structure, and specifically to security issues.

3. P2P And Security

Security expert Bruce Schneier is often quoted as claiming that Security is a process. As such, it matters very much whether security administration is centralized or decentralized. The basic model of the commercial Internet is the client-server relationship. As the network user base grew, less and less responsibility for administration was placed on the edges of the network, and more was concentrated in smart servers. This model is most evident on the World Wide Web, but file servers, mail servers and centralized security administration mechanisms such as Virtual Private Networks have all grown apace. In a centralized system, security policy can be dictated from a single location, and a that which is not permitted is forbidden policy can be enforced with firewalls, monitoring and intervention. On the other hand, centralization offers a single point of failure. Both direct malicious attacks and lax or negligent administration at the heart of a centralized network can undermine security for an entire system. With a single breach, outgoing or incoming content can be corrupted or intercepted, or the system can be rendered inoperable with a denial of service attack. Critical infrastructure systems such as the Domain Name System (DNS) have redundant implementation exactly because a single point of control is vulnerability in and of itself. In a decentralized P2P system, bad behavior has a locality impediment. Malicious attacks need to occur at or near every part of the system it wishes to affect. P2P systems lack the tools available to a centralized administrator, so 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. Malicious users can actively play the role of insidersi.e., they can run peers themselves, and often a great number of them. Doceur (2002) notes that this is incredibly hard for any open system to defend itself against this sort of attack, known as a Sybil attack (named after a famous Multiple-Personality Disorder case study). Several solutions are discussed below. As such, many systems are designed with strong security considerations in mind, depending on their function, from cover traffic to prevent monitoring of file transfer to a reputation system that can discourage single actors from controlling too many trusted nodes. As above, the wide range of security issues makes optimizing for all not a feasible option. This will become evident in our discussions of actual P2P systems below, but first we present some of the more common security demands. One way to think about P2P is whether it applies at the network level, the application level or the user level. At the network level, an adversary may try to break the routing system, or block access to information by impeding queries, or partitioning the network. At the application level, an adversary can attempt to corrupt or delete data stored in the system or in transit. Finally, the users themselves can be the subject of attack if an adversary goes after anonymity and privacy protections. Camp (2003) offers an operational definition of trust based on the risk that a system will fail to perform as specified. This is a broader view of security, but one that suits our purposes. In a P2P system, we are interested in not only how one aspect of the system behavior withstands attack, but how the entire system is defended. Since policy is not set dynamically, but is the sum of all behavior in the network, security also includes the expected behavior of other peers, and what their incentives are. Layers of protection interact. These levels translate into more specific security goals for system to operate reliably. In the examples that follow, we highlight security features of these systems, and the design choices made to offer optimal security for a given task. Discussions of cryptanalysis and other field-specific security issues employed by P2P systems are outside the scope of this chapter, as are more complete descriptions of these systems and their performance. Instead, these descriptions are intended to illustrate security mechanisms rather than fully describe all nuances of any P2P system. The case studies below highlight some of these goals. Secure distributed transport mechanisms like Tarzan ensure that the identity of the user and the content of the message remain a secret while secure overlay mechanisms are designed to ensure that the message gets through and gets there efficiently. Freenet and Free Haven ensure adequate file maintenance, so that users can reliably access any file they want, or any file that have been published, respectively. Groove allows networks of peers to manage themselves. Distributed storage mechanisms can create conditions for cooperative behavior and coordinated dependencies. Finally, we discuss accountability devices protect against free riding and nodes that are known to misbehave.

3.0.1. Subsubsection Heading Here.

Subsubsection text here.

4. Conclusion

The conclusion goes here. this is more of the conclusion

Acknowledgment

I would like to thanks to Mr. Mohd Zamri Murah for introducing Latex software in order to write a paper.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.