Category: IT Security & Ethics

Intrusion Detection Based on P2P Software

Zoltán Czirkos
Budapest University of Technology and Economics, Hungary

Gábor Hosszú
Budapest University of Technology and Economics, Hungary

INTRODUCTION

The importance of the network security problems come into prominence by the growth of the Internet. The article presents a new kind of software, which uses just the network, to protect the hosts and increase their security. The hosts running this software create an *Application Level Network* (ALN) over the Internet. Nodes connected to this ALN check their operating systems' log files to detect intrusion attempts. Information collected is then shared over the ALN to increase the security of all peers, which can then make the necessary protection steps by oneself.

The developed software is named *Komondor* (Czirkos, 2006), which is a famous Hungarian guard dog. The novelty of the system Komondor is that Komondor nodes of each host create a *Peer-To-Peer (P2P) overlay network*. Organization is automatic; it requires no user interaction. This network model ensures stability, which is important for quick and reliable communication between nodes. By this build-up, the system remains useful over the unstable network.

The use of the peer-to-peer network model for this purpose is new in principle. Test results proved its usefulness. With its aid, real intrusion attempts were blocked. This software is intended to mask the security holes of services provided by the host, not to repair them. For this it does not need to know about the security hole in detail. It can provide some protection in advance, but only if somewhere on the network an intrusion was already detected. It does not fix the security hole, but keeps the particular attacker from further activity.

BACKGROUND

The P2P networks comprise hundreds of thousands or millions of peers. That is why they are characterized by large dynamism, with a continuous process of nodes joining or leaving the P2P overlay.

Such large scale dynamism introduces several development problems. Neither a central authority nor a fixed communication topology can be employed to control the different components. Instead, a dynamically changing overlay topology is maintained and the maintenance is completely decentralized. The overlay is defined by links among nodes that are created and deleted based on the requirements of the particular application (Montresor, 2004).

Variability of P2P networks can be leveraged by implementing virtual networks based on super-peers. In the meantime, widely-used file-sharing systems such as Kazaa have applied the use of super-peers to enhance their search performance. In the field of the super-peer networks, the main focus is on centralized design of such networks (Yang & Garcia-Molina, 2003).

Until recently, most of the P2P applications deployed on the Internet had not any sophisticated mechanism for enforcing a particular overlay topology. The consequence of this was the adoption of simple communication models, such as flooding. Currently the situation has changed; many research projects have proved the importance of selecting, and proposed constructions and maintenance of appropriate topologies for robust P2P systems (Rowstron, & Druschel, 2001). Even popular file-sharing applications have started to consider more structured topologies (Kan, 2001). By introducing the concept of super-peer, their topologies are now organized through a two-level hierarchy. Nodes that are faster and/or more reliable than the ordinary nodes take on server-like responsibilities and provide services to a set of clients. A good example for this is the case of file sharing, where a super-peer builds an index of the files shared by its clients and participates in the search protocol on their behalf, leveraging them from participating in complicated protocols and reducing the overall traffic by forwarding queries only among super-peers.

The super-peer concept allows decentralized networks to run more efficiently by exploiting heterogeneity and distributing load to machines that can handle the burden. Also, it does not inherit the flaws of the client-server model, as it allows multiple, separate points of failure, increasing the robustness of the P2P network.

The applicability of the super-peer model is not limited to file-sharing, that is, it is possible to envisage distributed game systems (Smed, Kaukoranta, T., & Hakonen, 2003). In this case, multiple locations of a simulated virtual environment can be maintained by a distributed set of super-peers that control the virtual environment on behalf of their clients. Grid management systems and distributed storages are other

good possibilities for the usages of this architecture (Foster & Kesselman, 1999).

The construction and maintenance of a super-peer topology is, however, a difficult task. The extreme scale and dynamism call for robust and efficient protocols, capable to self-organize and self-repair a super-peer overlay in spite of both voluntary and unexpected events like joins, leaves and crashes. Another problem arises that in a P2P gaming, nodes must be clients of a single super-peer, and the sub-topology of super-peers must reflect the characteristics of the particular virtual environment that is simulated.

In order to avoid some problems that arise according to the super-peer model, Montresor (2004) proposed an enhanced protocol for the construction and management of super-peer-based overlay topologies. This protocol is based on the so-called gossip paradigm (Eugster, Guerraoui, Kermarrec, & Massoulié, 2003). In this case, every node periodically initiates an information exchange with a peer node selected randomly. The nodes involved in the exchange send each other information about their current status; whether they are super-peers or simply clients, the number of clients they are serving, and so on. Based on this information, role changes and/or client transfers can occur. A client may decide to become a super-peer and take responsibility for some of the clients of the other node to alleviate its load. Alternatively, a super-peer may decide to move all its clients to the other node and become a client by itself, to reduce the number of super-peers and, thus, the traffic generated by communication between super-peers.

THE SECURITY PROBLEMS OF P2P COMMUNICATION

The popular software, the *Instant Messaging* (IM) is the fastest increasing communications medium with an estimated 300 million consumer and enterprise IM users in 2005 (IMlogic, 2005). Global services such as AOL Instant Messenger, MSN Messenger, and Yahoo! Messenger each report over 1 billion messages sent per day, and IM traffic is expected to exceed email traffic by the end of 2006. As one of the most successful and widely-deployed applications on the Internet, IM has increasingly become the target for attackers to distribute IM-borne viruses, spyware software, worms, *SPam over IM* (SPIM), and malware attacks. Though widespread in adoption, IM is usually not protected and in user and corporate environments, leaving it vulnerable to attacks and exploits. These threats have grown exponentially over the past few years, increasing the need for real-time threat response for IM and P2P systems. As use of instant messaging clients and P2P networking increases, new viruses and other malware software are increasingly applying these mechanisms to disseminate.

Recently, the IM worms are more and more sophisticated and cross over from one network to another. In 2005, worms have been detected that are propagating over the Microsoft MSN public network and are crossing over into internal enterprise IM deployments, including Microsoft Live Communications Server environments (IMlogic, 2005). The growing prevalence of public to private network hopping by IM malware and worms will most likely increase in the near future, especially as IT organizations leverage public IM or connectivity to the public IM networks. The threats of multilingual worms are also growing, indicating a large sophistication for disseminating IM worms across geographic areas.

In order to monitor the network and to recognize the activity of the malware applications, a lot of different monitoring systems have been developed. One of them is the Netmon, which is a comprehensive network monitoring appliance that gives a complete perspective of the user's network (Netmon, 2005). Using Netmon, many kinds of malware can be identified, including worms, browser toolbars and plug-ins, and so forth. The Netmon can identify the activity of other types of network software, for example, P2P file sharing applications like KaZaA, E-Donkey and BitTorrent.

The Netmon apply the *Simple Network Management Protocol* (SNMP) to monitor the usage of many types of core network devices, including routers, gateway, firewalls, and switches. The implementation of the SNMP has two parts: one is the SNMP manager, and the other is the SNMP agent. While the SNMP manager is used by the network administrator, the SNMP agents are resided in the network devices to be monitored. The main task of the SNMP is extracting information from the *Management Information Bases* (MIB), which are maintained by the SNMP agents.

SNMP can also be applied for monitoring a lot of other kinds of devices, including network printers, hubs, and more. From the security viewpoint, the Netmon has an important feature, namely, its flexible port scanning tools. It can determine which ports are open for requests. The user is able to compare earlier scan results with the latest ones to spot newly opened ports.

TYPES OF PROTECTION

Security of a system can be increased with strict rules of usage. Applications doing this locally are Bastille-Linux (2006) and SeLinux (2006). These applications provide security mechanisms built in to the operating system or the kernel. Network security is also enhanced with these, but that is not the main purpose of them.