INTRUSION DETECTION AND RESPONSE TO AUTOMATED ATTACKS

An Overview of Automated Threats To Computer Infrastructure

Shawn Maschino

Graduate School of Computer and Information Sciences, Nova Southeastern University, 3301 College Ave, Fort Lauderdale, FL, 33314, United States of America

Keywords: Computer Security, Intrusion Detection, Worm, Virus, Denial-of-Service.

Abstract: This survey paper investigates current research in the fields of intrusion detection and response for automated attacks such as worms, denial-of-service, and distributed denial-of-service attacks. As the number of networked systems rise the ability to detect and respond to attacks is an essential part of system security for protecting data and ensuring availability of systems. This paper highlights current risk due to the latest automated attack technology and applies historical and current research to show the information security approach to detecting and preventing these types of attacks. Recent technologies such as virtualization and grid computing are discussed in relation to the roles they play in this area, and future areas of work are addressed.

1 INTRODUCTION

Intrusion detection research has been going on for over 20 years, and while much progress have been made new technologies and Internet growth make the need for updated intrusion detection and response methods a requirement in computer information security. With advances in the propagation rates of automated computer system attacks, as seen in Code Red, Nimda, Anig, and the SQL Slammer worms, there is elevated risk to computer systems.

In this paper the term "automated attacks" will be used to reference all scripted and self-propagating forms of computer attacks. These categories of attacks include worms, denial-of-service (DoS), and distributed denial-of-service (DDoS) attacks. These attacks are very common because they are easy to create and highly effective.

Worm-based attacks rely on security flaws on computer systems, such as unchecked buffer-overflow and format-string weaknesses. Worm writers exploit these weaknesses to run commands on a remote system, and then use the compromised systems to infect other computers. Many worms are coupled with a virus payload which would then damage the infected system or install a back-door which could be used to launch future attacks or steal data.

A recent example of a worm is the SQL Slammer worm, which infected 90% of vulnerable hosts, over 65,000 systems, in less than 10 minutes (Moore, Paxson, Savage, Shannon, Staniford, & Weaver, 2003). The damage from this worm alone was estimated at one billion dollars (Lemos, 2003). These costs include the time to repair the infected systems, patching efforts to secure vulnerable systems. and the loss of productivity due to down-time and slow network response.

The other category of automated attacks, denial-of-service and distributed denial-of-service attacks, are usually targeted at one specific system. While worms are written to spread quickly and infect systems with viruses or install backdoors or agents, DoS and DDoS attacks are used to stop systems from performing valid actions.

Denial-of-service attacks come from one host, and they overflow the target with multiple invalid requests so that the target cannot respond to valid requests, and therefore cannot provide the service it is intended to provide.

Distributed denial-of-service attacks work on the same principle, but rather than one host attacking a target there are many coordinated hosts performing the attack. This is done by having agents installed on many systems, usually put in place via a worm or Trojan virus. These agents can have a time and target built-in for when the attacker wishes the DDoS attack to begin, or a message can be sent out

Maschino S. (2005).
INTRUSION DETECTION AND RESPONSE TO AUTOMATED ATTACKS - An Overview of Automated Threats To Computer Infrastructure.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 522-525
DOI: 10.5220/0002540805220525

to these agents when the attacker wants to launch the DDoS attack.

2 INTRUSION DETECTION HISTORY AND OVERVIEW

In 1986 the first model of an automated system for intrusion detection was published (Denning, 1986). Denning's model focused on being able to detect an attempted break-in, successful unauthorized break-in, unauthorized actions by legitimate users, and virus, trojan, and denial-of-service attacks.

The most common intrusion detection systems (IDS) use methods based upon system and/or network monitoring implemented in software to detect attacks. These systems watch the activity of traffic on the networks, and the actions of the computer looking for any anomalies in usage. IDS software has a library of known attack signatures which identify common attacks or known worm and virus infection methods. If the IDS sees any activity which is a known attack it will report the attack and take action based upon the rules of the system.

The key to successful detection of an attack by an IDS system requires that the attack signature is known. Attacks that do not follow known attack patterns or signatures will not be detected by IDS systems. This weakness can be used by attackers who can evade detection by writing attacks that will not trigger an IDS alert.

In order to mitigate this weakness it is crucial to keep IDS signatures current. Once a new attack is found a new signature needs to be generated so the IDS systems can detect those types of attack. In order to create an IDS signature the following is required to be known:

- That there is an attack which uses set steps or patterns which can be described as a signature
- Any system state requirements in order for the attack to be usable
- The steps of the attack
- Any system state changes after a successful attack

The first point requiring knowledge of an attack is very important, as if that is unknown the rest of the items would be not be possible to identify. If the presence of an attack is known the IDS systems can identify vulnerable systems by knowing the system state requirements for the attack to take place. For example a vulnerable system may be one with a specific version of a piece of software running.

The most significant item for the creation of the signature is having the steps of the attack so that the IDS scanners can watch for these activities. In buffer-overflow attacks this would be the IDS looking for the correct length string of characters followed by the shell-code or other commands that are passed to exploit the vulnerability.

Even with known attack signatures intrusion-detection systems are far from 100% efficient and reliable. Current network speeds have a much higher bandwidth than most computer systems can process in real-time. Due to the fact that IDS are software implementations this means the IDS cannot process all network traffic, and the accuracy of the system will be limited by the amount of data it can capture and analyze.

This is where the fourth point mentioned above becomes significant. If an attack gets by an IDS scanner because it couldn't process all the traffic, it should be able to detect there was an attack if it finds the state of a system has changed to one that would indicate it has been compromised. At this point it would be too late to prevent the attack, but it can alert the system administrator and take a responsive action such as preventing the system from sending outgoing traffic therefore protecting any data that may be stored on it.

3 RECENT RESEARCH IN DETECTING AND RESPONDING TO AUTOMATED ATTACKS

For monitoring attacks there have been efforts by many groups to centralize the collection and analysis of network and system data for enhanced detection of security threats. In most cases this is done by the installation of agents on a large number of systems which collect firewall and system logs and report data back to a centralized source which processes the data to report out on worm propagation and virus infection.

The Internet Storm Center (ISC) is one the leading groups doing this type of analysis. Using the DShield Distributed Intrusion Detection system they collect firewall data from numerous sources on the Internet. In September 2004 the ISC processed over 1 billion logs for analysis (DShield, 2004). Using that data it is possible to query the ISC database to generate reports on traffic usage and worm propagation times. At the time of this writing the average time between worm attacks on an Internet host was once every 14 minutes (Internet Storm Center, 2004).

While this type of data is beneficial for finding trends of known attacks and network usage, it does not directly improve our ability to enhance system security.

To further improve the knowledge-base for securing systems there has been a lot of attention given to using honeypots to gather attack data. A honeypot is term used to identify a system which looks valid to attackers, but which was put in place to attract attacks so the activity can be monitored without them knowing they are being watched. Once these attacks are discovered they result in a better understanding of the evolution of attack methodologies, and they allow the creation of new signatures for intrusion detection systems.

One of the most popular honeypot configurations is the honeyd Open Source daemon. Using this daemon in a honeycomb configuration of systems can be used to automatically generate signatures of attacks for intrusion detection systems (Kreibich & Crowcroft, 2004).

Automatic IDS signature generation based on learned attacks from honeyd installations is quicker than the manual method of creating signatures, but it is flawed. The two biggest areas that need improvement in honeycombs and other similar tools are to minimize the number of invalid signatures that are created due to mistaking valid requests as attacks, and ways to improve the speed of processing requests.

Another current area of research being done to improve protection against automated attacks is advanced worm modelling using simulations to create early warning systems for worm propagation. In 2003, worm simulation and modelling was done based on a recursive algorithm using a Kalman filter (Zou, Gao, Gong, & Towsley, 2003) to investigate the increasing speed of worm propagation.

With the Kalman filter model a simulation found that the Slammer worm would have been able to infect 100% of vulnerable hosts on the Internet in 3 minutes if given unlimited bandwidth. Results show the first 1% of vulnerable servers were infected in 45 seconds. The only thing throttling the spread of the worm was the lack of network bandwidth, because the Internet's infrastructure could not carry the worm at the speed it was trying to spread.

Based on the speeds of recent worms we can see manual efforts to combat fast-spreading worms, such as OS and software patching, will not be enough to secure systems. In order to counter these types of attacks we need advanced detection and response systems that will automatically be able to close the weaknesses that the attacks use within minutes, if not seconds, of the first attack.

Tupakula and Varadharajan proposed an agent-based model in 2003 which would have agents on edge routers communicating with other routers within a LAN that would be able to detect this type of traffic. It would then determine the router which was closest to the source of the attack and would block the attacker's traffic at the point closest to its source so it would not flood the rest of the network.

Another agent-based model was proposed by Gorodetski, Kotenko, and Karsaev (2003) for intrusion detection and learning. Unlike the router-based model, these agents are installed on the computer systems and they communicate with each other to share captured data which was seen as an attack to learn and protect against new attacks.

While these types of methods have much potential in enhancing network security they are usually limited in usability as they require agents or new protocols to be used to be successful. Without computer industry agreement in which standards to use and a way to address the weaknesses in legacy systems we will still be limited to going without protection or having varied solutions put in place by individuals or organizations.

4 THE FUTURE OF INTRUSION DETECTION AND RESPONSE FOR AUTOMATED ATTACKS

Recent technologies, such as wireless networks, grid computing, and OS virtualization require investigation to ensure they are implemented to minimize possible infection by automated attacks.

Wireless networks pose a large risk to network security. The widely-used 802.11 wireless network standards offer poor built-in security and allow an easy place for attackers to connect to networks. With wireless networks it is easy for an attacker to launch attacks from a laptop while parked in a car outside a location with a wireless network. Once the attack is launched the attacker can easily move away, untraceable.

Another newer technology which is becoming widely used is grid computing. Grid computing allows users to access data and resources distributed across many systems as if they were located in one place. This can be highly efficient as processing power and resources are able to be balanced across many systems, but it also requires very specialized security to be used.

If grid computations and data distribution aren't seamless to users the usefulness of a grid would be minimal, but if all the systems blindly trusted each other it would be very easy for attackers to gain access to many systems by getting in to any system in the grid. It would also allow worms to spread very quickly if they weren't challenged once one grid system was infected.

Many grid computing farms use the Grid Security Infrastructure (GSI) model, which is based on Public Key Infrastructure (PKI) to give each user and resource a unique identifier. Users need to authenticate to access any grid resources, and resources ACLs are based on the unique identifier given to the users. However not all GSI grid systems are protected, as not all software running on grids are GSI-compatible, and the required key management can be cumbersome on large grids.

The final technology we will review in relation to intrusion detection and automated attack prevention is virtualization. Virtualization allows multiple operating systems to run on one piece of hardware without them knowing they are on a shared system. The OSes that are virtualized are known as guests, and the OS they run under is known as the host.

From a security perspective virtualization means that if a worm or attack can infect the host, it could take down the guests on the host making a DoS attack much more effective. If a DoS attack takes down a non-virtualized system only one OS is affected, but the number of affected systems would be much higher in a virtual infrastructure.

Virtualization technology could also be used to identify or respond to attacks. If the host layer of a virtualized system is secure the host could possibly identify and respond to attacks on the guests. If a guest OS was being attacked and was prevented from talking on the network the host may be able to respond to the attack without needing the guest active.

5 CONCLUSION

As we have seen there has been much research in the areas of intrusion detection and intrusion response. Much of this research has been focused on automated attacks such as worms, DDoS, and DoS attacks. As automated attack writers produce more efficient attacks, and the number of hosts on networks grow, the need for quick detection and response is crucial.

With worms being able to affect large number of hosts in seconds, and no current systems which can close down these vulnerabilities with such short notice, there is currently a technology gap between the attackers and those trying to protect systems.

Current research is closing this gap, and as new technologies are introduced into networks and computer systems there is opportunity for smarter and faster response to attacks. However these new technologies also bring with them new insecurities.

The process of using new technologies to prevent security issues being followed by newer exploits will most certainly continue indefinitely, but in the area of preventing and responding to automated attacks there is a strong need to improve intrusion detection and response technologies, which currently cannot process or respond to threats as quickly as attacks can be generated.

REFERENCES

Denning, D. (1986). An Intrusion-Detection Model. 1986 IEEE Symposium on Security and Privacy.

DShield. (October 2004). DShield Records Added Report. Retrieved October 17, 2004, from the DShield Web site: http://www.dshield.org.

Gorodetski, V., Kotenko, I. & Karsaev, O. (July 2003). Multi-agent technologies for computer network security. International Journal of Computer Systems Science & Engineering, Volume 18, Number 4. 191-200.

Internet Storm Center. (October 2004). Average Time Between Attacks: Survival Time. Retrieved October 17, 2004, from the Internet Storm Center Web site: http://isc.sans.org.

Kreibich, C. & Crowcroft, J. (January 2004). Honeycomb – Creating Intrusion Detection Signatures Using Honeypots. ACM SIGCOMM Computer Communications Review, Volume 34, Number 1. 51-56.

Lemos, R. (January 31, 2003). Counting The Cost of Slammer. CNet News. Retrieved October 16, 2004, from the CNet News Web site: http://surveys.cnet.com/2100-1001-982955.html.

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., & Weaver, N. (2003). Inside the Slammer Worm. IEEE Security & Privacy. 33-39.

Tupakula, U. & Varadharajan, V. (2003). A Practical Method to Counteract Denial of Service Attacks. Retrieved October 23, 2003 from the Conferences in Research Web site: http://crpit.com/confpapers/CRPITV16Tupakula.pdf.

Zou, C., Gao, L, Gong, W., & Towsley, D. (2003). Monitoring and Early Warning for Internet Worms. Proceedings of the CCC'03 Conference, 190-199.