ISSN NO: 2770-0003
Date of Publication: 30-05-2022

Honeypot-based intrusion detection system: A performance analysis

Gulomov Sherzod Rajaboyevich[1*], Salimova Husniya Rustamovna[2*]

[1*]PHD, Associate Professor, Head of the Department of "Information Security", Tashkent University of Information Technology named after Muhammad al-Khwarizmi, Uzbekistan

[2*] Master's degree, specialty "Information Security", Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

Abstract: Attacks on the internet keep on increasing and it causes harm to our security system. In order to minimize this threat, it is necessary to have a security system that has the ability to detect zero-day attacks and block them. “Honeypot is the proactive defense technology, in which resources placed in a network with the aim to observe and capture new attacks”. This paper proposes a honeypot-based model for intrusion detection system (IDS) to obtain the best useful data about the attacker. The ability and the limitations of Honeypots were tested and aspects of it that need to be improved were identified. In the future, we aim to use this trend for early prevention so that pre-emptive action is taken before any unexpected harm to our security system

Keywords: Honeypot, Handheld computers, Decision support systemshacking, security, forensic analysis of honeypots, network.

Introduction: First of all, honeypot forensics is used to study and understand a hacker strategy and his tools but not to prosecute him. This science is very time consuming and according to honeynet project members, one hour of hacker activity can lead to more than 40h of forensic work. The suggested approach is to work on a copy of the original victim, that way the analysis process can be repeated from the beginning without losing any important data. Forensic in computer science require a perfect knowledge of hacker techniques as well as how different software works in general. Forensic science is to find evidences to make researches on it and trying to find some details and answers from it. The forensic science branch that we are interested in our thesis is computer forensics which is the same definition of forensic science but this time electronic devices are involved with our researches. The necessary data is obtained from the devices, and forensic investigators make deeper examination on them. There are several roles and responsibilities for forensic investigation. Forensic investigation is done with first responders, investigators, technicians, evidence custodians, forensic examiners and forensic analysts. (Kipper G., (2007)). The different honeypots we studied offered us several log files that a forensic party can analyze. The most common file to study when we talk about network security is the .pcap file that most honeypots are generating. This file contains all the packets exchanged between the attacker and its target. It can be opened with Wireshark and allow the forensic to see what communication happened. This file can be huge in size but contains very important information. The difficulty here is to sort the relevant information. In the case of a honeypot, we assume that all traffic is suspicious thus any IP address not within our network must be analyzed. This make the sorting easier than on a production network where the attack is harder to detect. Another part of the forensic work is called reverse engineering. When a hacker successfully compromises a system, he will most likely upload one or more malware. Reverse engineering take a closer look at these malware by decompiling it and trying to understand what are their purposes and how they work. Again this technique is very time consuming but can allow the forensics team to identify new threats. Honeypot system In the computer network is very important for network security, especially related to applications involving various interests, there will be many things that can disrupt the stability of the computer network connection, whether related to hardware (physical security, power resources) and related to software (System, configuration, access system, etc.). Disruption of the system can occur due to accidental factors performed by the manager (human error), but not least also caused by a third party. Disturbances can include destruction, infiltration, theft of access rights, misuse of data or systems, to criminal acts through computer network applications. Security of the system should be done before the system is enabled. The use of the system should be done before the actual system is enabled. Overall.

Materials: Honeypots are mostly used by military, research and government organizations. They are capturing a huge amount of information. Their aim is to discover new threats and learn more about the Blackhat motives and techniques. The objective is to learn how to protect a system better, they do not bring any direct value to the security of an organization.

Methods: Honeypots can capture attacks and give information about the attack type and if needed, thanks to the logs, it is possible to see additional information about the attack. New attacks can be seen and new security solutions can be created by looking at them. More examinations can be obtained by looking at the type of the malicious behaviors. It helps to understand more attacks that may happen. Honeypots are not bulky in terms of capturing data. They are only dealing with the incoming malicious traffic. Therefore, the information that has been caught is not as much as the whole traffic. Focusing only on the malicious traffic makes the investigation far easier. Therefore, this makes honeypots very useful. For the only malicious traffic, there is no need for huge data storage. There is no need for new technology to maintain. Any computer can be used as a honeypot system. Thus, it does not cost additional budget to create such a system.

Results: We studied all level of interaction honeypots and configured them. As first level of interaction honeypot, we deployed Honeyd. We explained the logic behind it and installed it correctly. Our findings about Honeyd are; Honeyd is the most popular low interaction honeypot but its problem is its age. The project is opensource but part of it is outdated and nobody seems to upgrade it. On the other hand hacker tools are evolving, so identifying this honeypot is not hard. Honeyd is using an old version on Nmap fingerprint to create fake virtual operating systems so by using a newer version of Nmap, the fake operating systems will not be recognized and Nmap will detect that there is a problem. Another limitation of Honeyd is the scripts bound to the different ports. With a basic scan it is possible to find which ports are open but as soon as the attacker tries to actually connect on a port, he will realize the service is fake. For example the script used for a Web server, by connecting it using telnet, thew server should send back replies but nothing is happening. Another problem is one cannot understand if there is an incoming attack to the system or not. Because there is no such alarm system that can make you understand that there is an attack. Information gathering is not very smart either. As a result the hacker can understand quickly that there is something wrong with the target and will abort his attack. Even unprofessional intruders can compromise the honeypot without spending too much time on it. Because it is very popular and easy to use well known techniques such as Nmap. There is no additional approach needed for it. Our second step was to configure medium level interaction honeypot Nepenthes. We explained how it works and how we studied on it in implementation part. However, we found some problems with Nepenthes too. First of all, Nepenthes is for capturing malware over internet. It is mostly used for this aim. Thus, it must be implemented very rapidly since threats for users over internet are increasing dramatically day by day. Nepenthes could not keep up with new threats. As new threats are arriving and Nepenthes is not up to date, it will not be able to capture malware. Another problem comes from the shellcode. Shellcode manager should consider about shellcode and understand it. As new threats cannot be captured, new exploits cannot be captured either. Furthermore, as we are investigating the problems and security flaws in our experiment, there is an important security flaw in Nepenthes structure. Nepenthes do not have transport layer security. Transport layer security is a protocol that gives security for communications throughout the internet. We think it is a real problem for honeypot deployment.

Conclusion: We explained honeypot systems in detail, and implemented low interaction, middle interaction and high interaction honeypots at laboratory. Our goal was to understand their strategy and how they are working in order to lure intruders towards the system. We discovered their security flaws in order to help researchers and organizations. Several companies are using honeypot systems to protect the whole organization’s network security, and researchers are making academic experiments on them at schools. As we all know network security is very significant for all computer systems because any unprotected machine in a network can be compromised in any minute. One may lose all the secret and important data of a

ISSN NO: 2770-0003
Date of Publication: 30-05-2022

company, which can be a great loss, and it is also very dangerous that someone else knows your important personal information. Thus, we tried to find answers for honeypots' security using all interaction honeypots possible. Our main goal for our thesis was to see if honeypots are easy to hack and check if they are really isolated from other networks like a organization's network. When a honeypot is compromised, is it possible to reach other systems and compromise them too ? After the system is compromised, is it possible to track the hacker by using necessary forensic science tools ? How efficient are they ? As we stated in results and analysis part,we easily hacked all the honeypots that we used for our thesis. Especially, low interaction honeypot Honeyd can be hacked easily without too much effort. As we stated before, any amateur hacker can seize the system and also can see that it is a trap system. Therefore, Honeyd is not a good honeypot as its features are not efficient to fool the hacker. As Honeyd is a deamon, it is just simulating a operating system's services. So, it is not possible to a hacker to seize other systems using Honeyd. For the intruder, it will not take time to see that the system is not real, so he will not continue compromising it. He will leave the system. For forensic part, Honeyd's log was sufficient to see the actions of the hacker. Next part was to try Nepenthes as medium interaction honeypots. The result was quite similar. Thus,we came up with this conclusion: Low interaction honeypots and medium interaction honeypots are just simulating the services of a real system, because of that it is not possible to capture significant data from intruders. They are slightly different from each other but the main idea is the same. As they are not real operating systems , it is not risky to build them. There is no need to mention about further attacks. So, we moved on to the last level. After working low interaction and medium interaction honeypots, we decided to deploy high interaction honeypots. We studied on Honeywall. Even though it is time consuming and difficult, we managed to create a structure and worked on it. Our result were more interesting than before.High interaction honeypots are not virtualizing the system. They are real systems.So, it is very risky but the captured information is important. After deploying the implementation correctly, we successfully hacked the honeynet, but not Honeywall itself. It was the result we were looking for. As we stated in this paper, honeypot systems are still very new but are a great tool to identify cyber threats. The problem nowadays is that a very good hacker will most likely be able to understand when he is attacking a honeypot. Low interaction honeypots will be able to identify mostly automated attack and will hardly be able to understand new hacker method. On the other hand, high interaction systems are here to entrap the hacker and make him give away his techniques and tools to the forensic team. The network administrator implementing this kind of honeypot should make sure that the system is completely isolated 33 from the production network. This is the best defense if the hacker compromises the honeypot. Network security is not a path many students are taking but we see it as one of the most important topics when we speak about computing. We were curious about this subject and decided to write a thesis on that field. This work taught us a lot about the black hat and white hat community. It also gave us an idea how huge and complex the forensic work is. New threats are discovered everyday and the best way to stay protected is to always stay up to date. By doing this simple task, most attacks will not have any effect on the system. The problem nowadays is that people using pirated version of an operating system are contributing to botnets. Their system does not support critical updates and they are more sensitive to automated attacks. Nowadays, the implementation and development of honeypots are under control by network security expert. The weakness of this system is that it is not backed up by a clear legislation. Most of the work in the future should be about improving the laws about honeypots. The current laws about honeypots in most of the countries are not clear. There is a gap between the lawyers and the IT professionals. They should learn to cooperate with each other in order to clarify the legislation and give a clear answer about the legality of this technology. A lot of work should be done in the future to improve this situation. On a technical aspect, the main difficulty is to keep up with the new attacks. These days, it is not hard to detect a honeypot system, most of the work should focus on making this technology stealthier.

Literature:

1. Y. Yun, Y. Hongli and M. Jia, "Design of distributed honeypot system based on intrusion tracking", *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pp. 196-198, 2011.

ISSN NO: 2770-0003
Date of Publication: 30-05-2022

2. J.C. Chang and T. Vi-Lang, "Design of virtual honeynet collaboration system in existing security research networks", *2010 International Symposium on Communications and Information Technologies (ISCIT)*, pp. 798-803, 2010.
3. L. Li, H. Sun and Z. Zhang, *The Research and Design of Honeypot System Applied in the LAN Security in Beijing*, pp. 360-363, 2011.
4. L. J. Zhang, "Honeypot-based defense system research and design", *Computer Science and Information Technology 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pp. 466-470, 2009.
5. T. Holz and F. Raynal, "Detecting honeypots and other suspicious environments", *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop 2005. IAW '05.*, pp. 29-36, 2005.
6. T. Zhi-Hong, F. Bin-Xing and Y. Xiao-Chun, "An architecture for intrusion detection using honey pot", *Machine Learning and Cybernetics 2003 International Conference on*, vol. 2094, pp. 2096-2100, 2003.
7. I.Kuwatly, M. Sraj, Z. Al Masri and H. Artail, "A dynamic honeypot design for intrusion detection", *Pervasive Services 2004. ICPS 2004. IEEE/ACS International Conference on IEEE*, pp. 95-104, 2004.
8. A.Herrero, U. Zurutuza and E. Corchado, "A Neural-Visualization IDS for Honeynet Data", *International Journal of Neural Systems*, vol. 22, 2012.
9. D. Puthal, S. Nepal, R. Ranjan and J. Chen, "A Dynamic Key Length Based Approach for Real-Time Security Verification of Big Sensing Data Stream" in Web Information Systems Engineering-WISE, Springer International Publishing, pp. 93-108, 2015.
10. Y. Mai, R. Upadrashta, X. Su and J. Honeypot, "A java-based network deception tool with monitoring and intrusion detection" in , Las Vegas, NV, pp. 804-808, 2004.
11. D. Puthal, S. Nepal, R. Ranjan and J. Chen, "DPBSV-An Efficient and Secure Scheme for Big Sensing data Stream", *Tustcom/BigDataSE/ISPA2015 IEEE*, vol. 1, pp. 246-253.
12. R. Talabis, "Honeypots 101: A Brief History of HoneyPots", *The Philippine honeynet project*, 2002.
13. R. Baumann, "Honeyd-A low involvement Honeypot in Action", *Original published as part of the GCIA practical*, vol. 14, 2003.
14. X. Li and D. Liu, "Automatic scheme to construct Snort rules from honeypots data", *Journal of Systems Engineering and Electronics*, vol. 16, pp. 466-470, 2005.
15. H. Artail, H. Safa, M. Sraj, I. Kuwatly and Z. AI-Masri, "A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks", *Computers and Security*, vol. 25, pp. 274-288, 2006.
16. D. Dagon, X. Qin, O. Gu, W. Lee, J. Grizzard, J. Levine, et al., *Honey stat: Local worm detection using honeypots*, pp. 39-58, 2004.