Global Information Assurance Certification Paper

GIAC Security Essentials Certification (GSEC)
Practical Assignment
Version 1.4b

Fatal Attraction

Jilloo Bharucha
January 4th 2004

Abstract

A part of network security involves playing a game of hide and seek with the attackers, and playing with a skilled intruder is tricky business. If you are trying to learn more about the latest hacker techniques or simply defend your system against attackers, a honeypot is a good place to start. A honeypot is a deception tool – a machine or system - that is connected to a network to act as a bait for attackers by containing tempting, but false data. It imitates a real network, and is capable of tracking and logging details of the attack. Honeypots are effective because the attacker does not know when and where the honeypots are present and that their movements are being monitored.

This paper is written to introduce the concept of honeypots and explain in detail their key advantages. Since a honeypot is implemented with the primary purpose of capturing attacks, this paper also takes a look at understanding the threat. The paper goes on to explain why adding a honeypot to your security arsenal provides a big boost to overall network security.

Introduction

My computer at home is doing strange things. It's been three days since I last used it, but the hard disk is whirring away furiously and the lights on the modem are dancing excitedly. Fairly normal activity really, but slightly unusual timing given the recent lack of use. I double click on the icon showing connection to my ISP. The number of bytes being transferred is increasing rapidly. How unusual – what's going on here? I toy with the idea that someone has "hacked" into my PC, but that's ridiculous because I don't have any information that could possibly be of any interest to anyone. The number of bytes being transferred is still increasing. I decide to prod around a bit and launch Windows Explorer. Hang on – there's some mistake here – it shows that my hard disk is full. Now I know that's not true because the last time I used my PC I checked this and there was well over 20 gigabytes of space left. I shut down and restart my PC; maybe that will reset things back to normal. But no – the hard disk is definitely full. And there's a whole lot of new folders that never used to be there before. And a whole lot of new files and programs. How did they get there? Who put them there? How do I find out?
… and thus began my fascination with honeypots.

What is a honeypot?

Honeypots have been around for years and are finding increased use as a learning tool in the field of intrusion detection. While most traditional security mechanisms aim to keep attackers at bay by protecting anything of some value, honeypots serve a different purpose. In his book 'Tracking Hackers', Lance Spitzner [1] defines a honeypot as:

> "A security resource whose value lies in being probed, attacked or compromised"

This implies that a honeypot is a system specifically designed to be attacked or broken into. The aim of this is to gain insight into the attackers techniques, tools and motivations and also log and monitor the activities of the attacker. A honeypot has no production value, and should therefore not be generating or receiving any traffic. Thus, any traffic that it does receive is deemed to be of a suspicious nature and any time a connection is made, it is most likely an unauthorised intrusion of a hostile nature.

Lack of information about the enemy is one of the greatest challenges faced by the security community today. Skilled attackers are highly trained, experienced and take extreme steps to ensure that their activities remain undetected. Their presence is often difficult to identify as they use multiple systems to carry out an attack, and take care to remove all traces of their presence from the scene of the attack. As a result there is little information is available as to what the threats are, what tools are used and what the motivations of the attacker are.

Honeypots act as a tool to help us study our attackers. They act as a platform where we can record each step of an attacker and watch them in action from the start of an attack right until the end. By capturing, logging and monitoring their activities from the point they first contact a system, we can better understand how the attackers operate and what they are after. Such information is of immense value to the security professional looking at securing the system.

Understanding the threat

If you build a network, they will come…

It is inevitable. Every PC that is connected in some way or form to another PC is vulnerable to an attack. This is despite the fact that most networks run some sort of security software to regulate both their internal connections as well as connections to the internet. It's the price we pay to remain connected to the world. No matter how small or insignificant you think your network is, chances are it will almost definitely be attacked at some point in its life. Most people live behind a false sense of security believing that since their system has no data or significant information, it would be of

[1] Spitzner, Lance. "Honeypots: Tracking Hackers." Addison Wesley, 2002

no value to the attacker, so no one would want to hack into it. This misconception that systems that hold no value are safe from attacks is a dangerous one, and far from the truth. In reality, a ny and every system is vulnerable to an intrusion attack, and at some point has perhaps already been the victim of such an attack. Accessing and obtaining sensitive and valuable information on a machine is only one of numerous reasons why attacks take pla ce. Attackers can also gain immense use out of idle systems as long as they are connected. A co mpromised system can be used to launch an attack on another system and that system in turn to attack another system and so on. Attackers use this method to ma ke it difficult for investigators to trace down the source of the attack. Or, as in the case of my home PC (as mentioned in the introduction), the attacker could simply use the hard disk to store sensitive information that has been collected from elsewher e, to minimise his risk of exposure and possible legal implications of owning that data.

Statistics captured by a number of organisations demonstrate how prevalent the threat is. Findings include:

- The life expectancy of a default installation of RedHat 6.2 server is less than 72 hours [2]. The fastest time recorded for such a system to be compromised was within 15 minutes of being connected to the internet.
- A default installation of Windows 98 Desktop was compromised i n less than 24 hours [3].
- The year till 30 th September 2003 saw 114,855 sec urity reported incidents on CERT, a US federally funded security research institute [4].
- Towards the end of 2003, my home computer was scanned on avera ge seven times a day.
- Accor ding to a report publish ed i n 2002 by the Computer Security Institute, 90 percent of the 500 corporations, government agencies, financial institutions, medical institutions and universities su rveyed detected security breaches in their systems during the previous year [5]

The statistics are shocking. They were all obtained using basic, simple systems of little or no value, connected to the internet and acting as honeypots.

It is always important to know who the enemy is, what tools he uses, what targets he is after, and what strategy he is intending to employ. Un derstanding the people behind these attacks enables us to better identify the type of security systems we need to build. The hacking community can effectively be divided into three types of attackers: the Script Kiddies, the GreyHats and the BlackHats. All of them have a common aim – to gain control of a system they are not authorised to access. However, the methods they emplo y and their motives differ considerably. The biggest similarity is that they are very dangerous.

The first type of attacker, the Script Kiddies' goal is to get root access to the system in the easiest way possible. They usually want to compromise as many systems as possible with the least amount of effort. Most of the tools used by them are easily,

[2] Schneier, Bruce. " Honeypots and the Honeynet project ." 15 June 2001
[3] Honeynet Project. " Know Your enemy: Statistics ." 22 July 2001
[4] CERT URL: http://www.cert.org/stats/cert_stats.html
[5] Computer Security Institute. URL: http://www.gosci.com

and usually freely available and are mostly automated, requiring very little thought or interaction. They are usually not technologically adept enough to "crack" the software themselves and depend heavily on tools created by the more sophisticated attackers. They know the steps and commands required to operate a certain suite of tools to exploit known vulnerabilities, but very seldom have the programming and systems knowledge required to discover and exploit new weaknesses and develop new tools. Script Kiddies use automated probing tools to scan for targets and randomly select and attack a target with no biases towards software, location etc. Their reward lies in their ability to brag about the number of systems they have compromised and they use this to try to raise their status among the hacking community.

GreyHats fall somewhere in between the less knowledgeable Script Kiddies and the more technologically advanced BlackHats. They have the tools and the knowledge to attack sophisticated systems, but for the most part, respect the law. GreyHats experiment with the grey area that blurs the legal and the illegal as a means to develop their skills and learn more. They tend to shy away when they hit the border that crosses over into the illegal.

BlackHats are perhaps the most respected of the hacker community. They are the ones who largely responsible for the public opinion of the notorious "hacker". Though less common, and representing only a tiny fraction of the population, Blackhats are highly experienced and very knowledgeable. They target relatively few systems, but ones of high value, and it is not unusual for an experienced BlackHat to spend months perfecting a single attack. They are usually financially or politically motivated, and the majority of successful known attacks have been against banks and financial organisations, the governments and military of various countries. BlackHats are also more difficult to detect because they take extreme measures to cover their tracks and often use multiple systems for a single attack, making it more difficult to be discovered. They use specialised tools designed for a specific attack, and often even develop their own tools which they seldom share with others. Very little is actually known about the advanced BlackHat because they are obsessively adverse to publicity even within the BlackHat community.

Advantages of honeypots

Honeypots play a supporting but important role in overall network security. It is certainly not enough to employ a honeypot as the only security mechanism in your network: however, honeypots work well with other security technologies, and have several advantages. These are discussed in detail below:

Deter, divert and confuse attackers

It is not surprising that experienced attackers are always worried about getting into trouble with the law. The more serious the attack, the more severe are the legal implications of being discovered. Few intruders will invade a network that they know is designed to monitor and capture their activity in detail. If an attacker knows your organisation is using honeypots, but does not know which systems are legitimate computers and which systems are honeypots, they may decide not to attack your

organisation because they are concerned about being caught or co mpromising their arsenal.

Honeypots also have the ability to provide defence through deception. Attackers can never be sure if they are co mpromising a real system or wasting their time on a honeypot. The false information that honeypots provide the attacker can confuse them. The idea is to make the attacker waste his time and bind his resources interacting with honeypots while keeping production server s free from any harm and monitoring all the attack activity at the same time.

The Deception Toolkit (DTK) is an example of a honeypot that uses deception to counter attacks. The DTK is a straightforward tool which generates fake information relating to a machine. It consists of a collection of scripts that emulate known vulnerabilities on a system, but the attacker cannot discover this from an innocent scan. They must actually attempt to exploit the vulnerability, and when they do, the DTK logs the a ctions. For example, an attacker scanning port 23 for a telnet server might encounter the DTK which would respond to the scan by providing the attacker with a realistic looking login prompt for a telnet server followed by a password prompt. If the attack er guesses the correct password, and gains access, the DTK would display a fake shell prompt, but the attacker would not know this because it looks identical to the real telnet prompt. The DTK is even configured to allow the attacker to carry out simple c ommands such as listing files in a directory, and viewing the password file, but none of the information displayed actually exists. While the attacker thinks they have found an exploit, the DTK silently records and logs all activity.

Enhance the IDS

Intrusion Detection Systems (IDS) are used to detect and alert on possible malicious events within a network. They work by passively monitoring all network traffic for unauthorised or suspicious activity. When such activity is identified, an alert is genera ted. IDS sensors are placed strategically at various points in the network where attacks are likely. These include the interfaces between the internal network and the internet and at critical points within the internal network.

The use of an IDS as a n etwork security device has its fair share of shortcomings. Network IDSs need to be extremely efficient at dealing with considerably high amounts of network traffic, which they must process in a timely manner. As network traffic increases over time, it is getting increasingly difficult to strike the balance between acceptable performance and acceptable packet loss. IDSs also have difficulty with false positives and false negatives. A false positive is an alert generated due to normal acceptable activity. They are likened to the “boy who cried wolf” scenario and are a waste of valuable time and resources. The danger is that if an IDS repeatedly generates false positives then system administrators begin to pay less attention to them and could miss out on an actual attack. A false negative is the opposite of a false positive, and occurs when an IDS fails to alert on a legitimate attack.

The use of a honeypot within a network can provide an additional layer of network

security along with the IDS. Since honeypots and IDSs are different in many ways, honeypots can complement an IDS to help overcome some of its shortcomings. Comparatively, honeypots collect smaller sets of data as they only log connections when other systems are communicating with them. This reduces the man hours spent in monitoring, managing and analysing the data. Honeypots also help reduce false positives and capture false negatives. Since all activity to and from a honeypot is by its nature deemed as suspect, false alerts are almost completely eliminated. Honeypots also capture false negatives since they log information on all activity with themselves including new attacks. They are a simple, cost-effective way to enhance existing IDS systems in a network.

Detect insider attacks

While most networks are well prepared for attacks from the outside, the danger of attacks from within are often overlooked, if not ignored altogether. System administrators and security personnel spend most of their time, money and resources ensuring that the network perimeter is sealed as best as possible and live under the false feeling of safety that if they protect themselves from the outsider, they are safe. Attacks launched from within an organisation are potentially more severe and aggressive. This is because an inside employee is trusted with special insider knowledge and information and knows what the prime assets of a company are. An insider knows exactly what is valuable, why it is valuable, how it can be destroyed and where to look for it. In addition, internal employees have ample time to browse around the systems and plan an attack. They can use the legitimate access that they already possess to gain additional unauthorised access to the systems.

Most IDS systems have difficulty detecting insider attacks. This is usually because organisations do not monitor the inside as heavily as the outside. Also, as the line between internal and external becomes increasingly blurred by corporate mergers and partnerships, an attacker can seamlessly move from one system to another making it difficult for the IDS to identify the origin of an attack. In addition, IDS's follow a rule based system, which can sometimes be misleading as a rule categorised as not suspicious due to the fact that a packet originated from within the organisation may actually be malicious and indicative of an attack, but an IDS is not intuitive enough to pick this up.

Internal users also often compromise security by installing Peer to Peer (P2P) file sharing applications that silently enable the sharing of the entire hard drive. There is also the added danger that these programs operate by trying to bypass the firewall they are situated behind. This creates an open door for attackers as it provides a direct access route into the network without having to go through a firewall. Once an attacker has access to one PC in a network, gaining access to others is only a matter of time. Most non technical users may be unaware that they are creating a security hole by doing this, and not just endangering themselves, but the entire organisation as well.

Honeypots can provide valuable information on the patterns used by insiders. Since packets are not meant to be either received or sent by a honeypot, any such activity

can be deemed as unusual. When placed within the internal network of an organisation, any activity generated on the honeypot is assumed to be from the inside. This may be either an internal employee or an outside attacker who has somehow managed to gain access to an internal machine, for example, using P2P as mentioned earlier. In his article "Don't ignore the threat from within"[6], George Lawton relates an incident where a honeypot was deployed with the aim of trying to monitor and capture internal attackers. The surprise was finding the companies chief operations officer trying to break in. Such an attack that would probably go undetected using conventional security mechanisms could prove to be dangerous for a company.

Defend against worms

At 11:34 am Pacific time on August 11, 2003, Microsoft began investigating a worm known as MSBlast which affected millions of computers connected to the internet running Microsoft Windows[7]. Symptoms included the system rebooting every few minutes without user input or systems becoming unresponsive. Code Red and Nimda worms are examples of two other worms that have attacked computers on the internet in the past. Worms work by infecting a host, and then using the infected host to search for more victims to propagate to and repeat the process on the new host. Each of these automated worms has been able to reproduce itself to every available system on the internet, resulting in a widespread security problem. Worms use localised scanning to propagate, the basis of which is to try and infect machines within close proximity to the currently infected machine.

The infection pattern for code red is as follows: 3/8 th of the time it attempts to infect a machine within its own class B address space (/16 network), ½ the time it tries to infect a machine within its own class A address space (/8 network) and 1/8 th of the time it would choose a random address from across the entire internet. Localised scanning appeared to be successful for the code red II worm. It allowed this worn to spread quickly within parts of the internet that had a high concentration of vulnerable hosts. This strategy allows a worm to spread very quickly within an internal network after it has already bypassed any external firewall or IDS system. [8]

A well configured honey net (an entire network of honeypot systems) is good at detecting worms that use localised scanning to propagate. By its very nature, any traffic to and from a honeypot is considered suspicious. Repeated scans for a specific port across the honeynet is indicative of an infected machine looking for a vulnerability. By analysing the data collected, it is fairly accurate to assume that scans that have occurred in under one second across numerous systems on the honeynet are most likely automated worm type exploits.[9] This information can be used to warn system administrators of an attack and allow them to take remedial action to by helping them develop a new signature to prevent the worm from propagating into the enterprise network.

---

[6] Lawton, George " Don't Ignore the threat from within." 14 June 2000
[7] Microsoft URL: http:// www.microsoft.com/security/incident/blast.asp
[8,9] Levine, John; LaBella, Richard; Owen, Henry; Contis, Didier; Culver, Brian. "The Use of Honeypots to Detect Exploited Systems Across Large Enterprise Networks." June 2003

Specially configured Honey pots are also instrumental in slowing down the attack by a worm, and affecting their propagation rate, potentially even stopping them. Called a “sticky” honeypot or “tarpit”, it causes the machine at the other end to get “stuck” for long periods of time. This solution works by taking unused IP addresses on the network and creating virtual machines that allow TCP connections to be accepted by a tarpitted port, but not allowing the connection to get back out. When probed by such scanning activity, these hon eypots hold the connection open, and ignore any requests by the attacking machine to close the connection. This means that until the connection times out, resources of the attacking machine are tied up. This is excellent for slowing down and preventing th e spread of a worm. One example of a sticky honeypot is LaBrea Tarpit which was designed as a response to the Code Red wor m.

Small sets of data of high value

Large amounts of data are generated and collected by security mechanisms in a network on a dai ly basis. The bulk of this data comprises of logs (system, firewall, database, access etc) and intrusion detection alerts. This data consists of some useful information but it is mixed with large amounts of legitimate traffic and system data, making it d ifficult to separate the useful from the useless, and derive much value from it.

Honeypots only capture suspicious activity, as any interaction with a honeypot is usually unauthorised and considered to be malicious. Instead of generating thousands of a lerts a day, they only generate a few, and instead of logging gigabytes of data a day, they only collect a few hundred megabytes. Honeypots tend to reduce much of the “noise” by collecting only small sets of data, but since this data almost certainly re lates to an attack, it is considered to be of a higher quality and value. No matter how good one is at parsing and analysing endless log files, dealing with a smaller amount of data makes the task much easier and reduces the chances of things being overlook ed.

The key to effective data collection is to collect as much data from as many sources as possible without the attackers knowledge. A good place to collect data would be from the honeypots firewall. All firewalls have the ability to examine and log a ny traffic passing through them, without the attackers knowledge. Firewalls reduce risks by scanning and filtering both incoming and outgoing traffic for malicious content or potential exploits. They also keep records of both successful ad blocked networ k traffic including their source and destination IP addresses, the date and time of an attack, and packet header information.

Another data collection tool is the Intrusion Detection System (IDS). An IDS monitors a system or network for malicious activity. One of the key features of an IDS is that it can capture every single packet that traverses a network. While this is a useful feature, it is usually not feasible in most enterprise networks as the quantity of data captured would be enormous! However, s ince honeypots usually have very little activity, deploying an IDS can ensure that all network activity is recorded and stored, and can be used for further analysis. They will provide you with a key by key view of what the attacker does and sees.

The honeypots own system logs also act as a tool for data capture. Almost all available honeypot software has a mechanism for logging and storing details of activities captured by it. However, since these logs will be the intruders first target if he suspects he is being tracked, they are extremely susceptible to alteration. It is therefore vital that these logs are automatically duplicated to a remote system so that even if the logs are modified or destroyed on one system, there is always copy on another.

The data collected by a honeypot is considered to be of a high value because it leads to a better understanding and knowledge about the attackers, which in turn can help to increase overall network security.

Simplicity

The simplicity of honeypots is perhaps their biggest advantage. Most security mechanisms require a lot of time and effort to be spent on initial configuration before they can be deployed into a production environment. Firewalls enforce policy and require a set of rules to be defined. These rules are detailed and technical and often need to be written by a well trained and experienced firewall administrator. Since these rules are also likely to change from time to time, and new rules need to be added and updated, maintenance of firewalls is a time consuming task. IDS systems also usually require a complicated initial set-up since they follow a method known as signature based detection. This is based on the premise that malicious or abnormal network traffic fits a distinct structural pattern whereas normal traffic does not. It is therefore possible to create an attack signature for malicious traffic based on its content and structure. A rule can then be developed based on this signature and stored in the IDS's rules engine. Another rule is then configured to generate an alert to inform the administrator whenever any traffic that matches the signature is detected in the network. Building and maintaining this rules base is a time intensive task, and even the slightest error in configuring these complicated signatures can cause the IDS to miss an attack.

Honeypots do not require a complex rules base to be configured, neither do they require lengthy signature databases to be built and maintained. They follow a simple "plug and play" mechanism whereby they can be connected into any part of any network and have the ability to start reacting immediately. While there are some honeypots that can me more complex, even the simplest ones are extremely effective, reliable and quick

In addition, honeypots do not require any expensive hardware, and can be built on minimal resources. This means that an old pentium computer with 128MB of ram can be configured to act as a honeypot that can easily handle an entire class B network.

Education and Research

Honeypots are invaluable in the field of security education and research as they provide us with data on the methods used to attack systems. This knowledge is extremely useful to designers of systems as they provide them with insight into the tools and motives of the

attacker. Available information is limited, and usually based on the speculations and assumptions of security personnel. Honeypots provide a first hand look at the actual attack, and log the attack for future reference.

Honeypots a lso have the capability to deploy a specific operating system on them that security personnel are concerned about being exploited. Say, for example that the security personnel are concerned about the safety of a SQL server. If they configure a honeypot to act as a SQL server that matches the original configuration, then they can monitor the honeypot for attacks to potential vulnerabilities. Any suspected compromises will then be reported and make the administrators aware of any fixes or patches required.

Resource Utilisation

Due to the high amount of traffic traversing a network on a daily basis, it not unusual for security mechanisms to occasionally fail because of resource overload. When this happens, the mechanism can no longer provide secure network monitoring and could potentially miss a significant attack. For example, when the buffer on an IDS sensor becomes full, it will start dropping packets. When gigabytes of traffic flow across the network, IDS sensors have difficulty capturing and monitorin g every single packet because the speed and volume of the traffic is too much for the sensor. Once the sensors buffer is exceeded, it's quality of service drops and it can miss a packet that contains information about a potential attack. Or a firewall may no longer be able to monitor traffic passing through it because its connection tables are full and it is running low on resources. In this case, the firewall which normally blocks only unauthorised activity responds by blocking all traffic to and from it until the resources are re leased. Resource overload therefore has significant security implications because it prevents such mechanisms from functioning correctly.

Honeypots on the other hand encounter comparatively little traffic. They do not capture all activity on a network, but rather only activity directed specifically at itself. They therefore usually do not face the problem of resource exhaustion or overload, and are able to continue their capturing and logging activity with consistency and ease.

Capture new tools and tactics

Firewalls and IDS systems are configured with know signatures and principle based rules, and are designed to raise an alert based on these. A new exploit which does not have such a signature or rule already written for it will usually go undetected. Honeypots are designed to capture anything thrown at them, including tools or tactics never seen before. They are therefore quicker to raise an alert about any new suspicious activity. They are also capable of capturing details of the exploit. One such exploit was encountered by the Honeynet Project, on 8 th January 2002. Researchers observed that one of the servers in their honeynet running an unpatched version of Solaris8 Sparc was remotely compromised by an unidentif ied exploit. The exploit used the CDE (Common Desktop Environment) Subprocess Control Service (dtspcd) to cause a buffer overflow. This was the first attack of its

kind in the security community, and the details captured by the Honeynet Project resulted in a CERT advisory [10] which made the attack known to the security community.

In depth information logging

Security professionals often question what the best way to track an attacker is without the attacker knowing it. The best solution is provided by multiple logging or layers. In a honeypot logging needs to be as silent as possible. If an attacker suspects that his activities are being logged, he will often try to take some action to erase these records. Since a single layer of logging can easily be altered or deleted, it is best not to depend only on that, and instead employ multiple logging positions. Different logging views also provide better understanding on what the attacker is trying to do. It is also important that the integrity of logs can be guaranteed. A good logging practice is to create an unaltered copy of the logs on a system the intruder cannot access, as well as the honeypot itself.

Information gathering on a honeypot is mostly passive. The honeypot accepts all incoming packets, but does not query the third party for specific information about itself. Additional information about the attacker could prove valuable when analysing an attack. It is possible to get more information about a person, an IP address or an attack by querying specific services or machines. This can be very powerful as valuable information can be found. However this attempt is also dangerous as the attacker could take notice and vanish. The following services are availabe to query third parties: whois, finger printing network traffic, portscan, finger. [11]

Logging can further be improved by using a network sniffer on the honeypot that has the ability to capture all keystrokes made by the attacker as well as screen captures to see what the intruder sees.

Return on investment (cost effectiveness)

Unfortunately, management, who hold the strings to the money purse are usually not technically inclined. They often question the value of an investment in terms of the benefit received. It is sad but true that management sometimes begins to question the return on an investment when they perceive that there is no longer a threat. For example, management may question the investment in an expensive firewall because the organisation has never been hacked. What they do not realise is that they have not been hacked because the firewall prevented this in the first place. Security managers are often asked to justify a spend on any new technology or systems. They often use firewall and IDS logs to demonstrate potential attacks that have been prevented. But until a system is actually compromised, management do not really believe they are at risk, and by then it is often too late to mend the damage.

[10] CERT URL: http://www.cert.org/advisories/CA-2002-01.html

[11] Baumann, Reto and Plattner, Christian. "White paper: Honeypots." 26 February 2002 URL: http://www.security.rbaumann.net/download/whitepaper.pdf

Honeypots, on the other hand can quickly and effectively demonstrate their value. They are quick to justify their own spend by capturing all unauthorised activity. Whenever they are attacked, we know that the bad guys are out there, and that the next attack can be of a more serious nature. Presenting management with a list of attacks that have actually taken place holds more value than presenting them with a list of logs of potential attacks. They are more likely to support investments in security technologies if they genuinely believe that they may be the victim of an attack. Lance Spitzner in his book "Tracking Hackers" describes a situation where he was once asked to do a presentation to the Board of Directors of a large financial organisation on the state of their security. About half an hour before the presentation he connected to the network to make some last minute changes to his presentation. At the time he had a honeypot running silently in the background of his computer, and it captured a scan, probe and attack on his computer. He managed to win over the Boards support by demonstrating the attack, keystroke by keystroke that active threats not only existed, but tried and succeeded in penetrating their network.

Support for Encryption and Ipv6

The move towards encryption is increasing, as organisations are increasingly adopting encryption mechanisms such as IPSec and SSH. However, most existing security mechanisms such as firewalls and IDS systems depend on being able to view the contents of a packet to provide security. If encryption is employed by an organisation, all that can be seen by the IDS is encrypted packets on the network, which it cannot understand, and therefore cannot respond to.

Ipv6 is the new version of IP (Internet Protocol). The version of IP currently in use is Ipv4. At the moment IPv6 is still relatively new and not widely adopted. Most IDSs are not capable of analysing or understanding Ipv6 packets because they cannot decode the data correctly.

Unlike most existing security technologies, such as IDS systems, honeypots work well in encrypted or Ipv6 environments. It does not matter what the attackers throw at the honeypot, the honeypot will detect and capture it. Even if an attack is encrypted, the honeypot will capture the activity without them knowing it. This is done by inserting kernel modules on the honeypot system that captures the attackers actions. A honeypots ability to log activity is unbiased to the IP protocol used. It does not matter whether an attack uses IPv4 or IPv6. In one documented case, a Solaris honeypot detected and captured an attack where attackers attempted to hide their communications using IPv6 tunnelling within IPv4. [12]

Conclusion

The purpose of this paper was to demonstrate the usefulness of honeypots as a tool contributing to the overall security of a network, be it an enterprise network or a personal one. Honeypots are attractive to both attackers as well as security professionals, and are being deployed more frequently. As honeypot technologies become more popular, attackers are getting increasingly aware of the trap that lies in

[12] Spitzner, Lance. "Honeypots: Simple, Cost-Effective Detection." 30 April 2003

store for them, and are developing their own methods to detect such systems. The more we learn about the tricks of the attackers, the more security we can build into our systems.

My computer at home is still doing strange th ings. It's been a few days since I last used it, but the hard disk is still whirring away furiously and the lights on the modem are still dancing excitedly. I know that someone has hacked into my PC…… because this time they are being watched.

Reference s

Anuzis, Michael. " Incident Analysis of a Compromised OpenBSD 3.0 Honeypot ". July 2002
URL: http://www.anuzisnetworking.com/whitepapers/obsd30/

Baumann, Reto and Plattner, Christian. " White paper: Honeypots ." 26 February 2002
URL: http://www.security.rbaumann.net/download/whitepaper.pdf

Chuvakin, Anton. " Lessons of the Honeypot I: Aggressive and Careless ." 19 June 2002
URL: http://www.infosecnews.com/opinion/2004/06/19_04.htm

Cooper, Mark. " Baby Steps with a Honeypot ." April 2002
URL: http://www.mh c-online.co .uk/babysteps.htm

Evans, Andrew. " Honeypots – Weighing up the Costs and Benefits ". 28 October 2002

Honeynet Project. " Know Your Enemy: Worms at War ". 8 November 2000
URL: http://project.hon eynet.org/papers/worm/

Honeynet Project. " Know Your Enemy: Honeynets ". 26 April 2001
URL: http://project.honeynet.org/papers/honeynet/

Honeynet Project. " Know Your Enemy: A Forensic Analysis ". 23 May 2000
URL: http://www.project.honeynet.org/forensics

Koziol, Jack. Intrusion Detection with Snort . Indianapolis, Indiana: Sams Publishing, 2003

Lawton, George " Don't Ignore the threat from with in." 14 June 2000
URL: http://www.itworld.com/sec/2052/ITW1124

Levine, John; LaBella, Richard; Owen, Henry; Contis, Didier; Culver, Brian. " The Use of Honeypots to Detect Exploited Systems Across Larg e Enterprise Networks ." June 2003

Liston, Tom. " LaBrea – The Tarpit ." October 2000

Miller, Toby. " Intelligence Gathering: Watching a Honeypot at Work ". 10 January 2003
URL: http://www.securityfocus.co m/infocus/1656

Oudot, Laurent. " Fighting Internet worms with Honeypots ." 23 October 2003
URL: http://www.securityfocus.com/printable/infocus/1740

Raikow, David. " Building your own Honeypot ". Zdnet, November 2000

Schneier, Bruce. " Honeypots and the Honeynet project ." 15 June 2001
URL: http://www.schneier.com/crypto -gram-0106.html

Spitzner, Lance. " Honeypots: Definitions and Value of Honeypots ." 29 May 2003
URL: http://www.tracking -hackers.com/pap ers/honeypots.html

Spitzner, Lance. Honeypots: Tracking Hackers . Addison-Wesley, 2002

Spitzner, Lance. " Honeypots: are they Illegal? ." 12 June 2003
URL: http://www.securityfocus.com/infocus/1703

Werrett, Jonathan. " Implementing and testing an Intrusion Detection Honeypot ." 2 June 2003

William, Martin. " Honey Pots and Honey Nets – Security through Deception ". 25 May 2001

URL: http://www.all.net/dtk

URL: http://www.project.honeyneet.org/project.html

URL: http://www.pcwebopedia.com/term/s/script_kiddie.html

URL: http://www.recourse.com/products/mantrap/trap.html

URL: http://www.hackbust ers.net/LaBrea

URL: http://www.gosci.com

URL: http://www.microsoft.com/security

URL: http://www.cert.org

URL: http://www.securityfocus.com