Deception: Technologies and Strategy for Cybersecurity

Abdulrahman Yarali, Faris George Sahawneh
Institute of Engineering, Telecommunications Systems Management
Murray State University
Murray, KY, USA
ayarali@murraystate.edu, fsahawneh@murraystate.edu

Abstract— **There are many different types of security threats to individual, business and government enterprises. Although there are efforts to ensure that technologies work to the best levels, there are still several challenges. Keeping the security of the enterprise system in view, security designers have introduced multiple security perimeters like firewalls and threat detection and response (TDR) systems. These frameworks directly contribute to detecting and defending any unwanted action into the system. The objective of this paper is to survey the role and strategies of deception in cyberattack. A detailed discussion of technologies and tools for combating unwanted adversary actions are presented.**

Keywords- ***deception***; ***cybercrime***; ***cyberdefence***; ***cybersecurity***; ***honeypot***; ***honeynet***; ***adversarial mind manipulation***

I. Introduction

It is necessary for enterprises to find ways to detect and defend human attackers in real time. Deception tools and technologies are the best solution to the problem. Deception tools work in real time to detect and defend unwanted actions from human attackers that attempt to steal critical information from the system. The tools are very accurate and automated, which keeps a firm eye on the malicious activities happening in the system. These tools can track the activities from the internal network that cannot be tracked by other cyberdefence frameworks. Deception tools operate proactively, working to deceive the attackers. The tools are designed to think like cyberattackers, which enables them to proactively defend and defeat the intruders. The tools include threat intelligence systems which can differentiate among the potentially harmful threat alerts and normal alerts.

II. Evolution of Deception Technologies

Cybersecurity has been a major concern for enterprises for many decades. For much of the 70s and 80s, threats to computer security were clear and present. In 1986, the German computer hacker, Marcus Hess, hacked an internet gateway in Berkeley and used that connection to piggyback on the ARPANET. Continuous malicious activities conducted by the adversaries have always caused inconvenience for cybersecurity implementers. The initial silo networks were very general and prone to attack. There was no mechanism to save the network from cyberattack. As time passed, the technology evolved, and the attacks became very common, which raised concern for the security implementers. Initial researchers helped the security teams to make different types of firewalls that blocked the adversaries from entering the network. These security measures worked for some time until the attackers learned to bypass those firewalls. The security researchers then tried to find out other ways to defeat the attackers, and they created deception technologies [1].

The basic idea behind deception technologies is to trap the adversaries and confuse actual assets and dummy assets. Security experts created dummy layers of the network along with the actual layer of the network as a deception for the attackers. Many attackers fell into these deceptive layers at first, but later found ways to ignore the deception technologies. The security implementers are now trying to create more transparent and effective deceptive technologies which can make the adversaries more suspicious about the network assets.

The malware was easily detectable in the past. The attackers didn't know about the potential ways in which this malware could be transformed. Today, malware is more advanced and sophisticated. The attackers have changed malware through packing, polymorphism, and encryption. These operations not only make the malware more sophisticated but also more undetectable. The previous security mechanisms were designed to respond to threats and malware in a fixed period of time. The continuous and constant stream of malware now blocks the detection system and enters the internal network very easily. The zero-day threat is an example of such sophisticated malware which can easily avoid any antivirus and enter the system without being detected. The threat detection and response (TDR) system helps to detect and defeat the potential attackers. The framework defends the system from many types of harmful malware [2].

With the passage of time, malware has become more sophisticated, and human attackers have become more

978-1-7281-5506-7/19/$31.00 ©2019 IEEE
DOI 10.1109/SmartCloud.2019.00029

advanced. Attackers have been able to avoid the TDR system easily, which has created a problem for the security and risk management teams of the enterprises. The best solutions are deception technologies which are proactive and handle the threats and malware effectively. The security implementers are designing frameworks which incorporate the deception technologies with the TDR mechanism. The TDR provides low-quality data in terms of new and sophisticated malware attacks. This restricts the security frameworks to avoiding and defeating the malware. Deception technologies, along with the TDR frameworks, help to get low false-positive and high-quality data.

III. Deception Tools as Paradigm Shift

Deception tools have shifted the paradigm by changing "big data" into "useful data." The security techniques and tools in the past usually included firewalls and intrusion detection systems (IDSs). These systems worked well, but in the case of larger networks where the daily traffic is in the millions, these networks failed badly. For example, a normal server of a large enterprise gets millions of hits every day from different areas. It is nearly impossible to identify the attacker among these millions. Deception tools were not the direct and only solution proposed by the security researchers after finding this issue. The big data were proposed as a solution. In this situation, the traffic from the normal servers is monitored, and the suspected traffic is recorded. The recorded traffic is then studied to find out who the actual hacker is and what damage he has done.

This solution worked on a smaller scale. When the daily traffic crossed a normal limit, the recorded data grew and changed into big data. The enterprises began to employ big data experts to implement techniques like data mining to discover useful data among bulk data. The use of big data increased so much that it became nearly impossible to pinpoint useful data among the bulk. Deception technology created a paradigm shift because it changed the traditional trend of using big data [3]. It transformed the big data into useful data by reducing it to a minimal level. This greatly downsized the problem. Enterprises now do not have to dig through millions of entries to find out who attacked their system. They can directly find the hacker by looking into the traffic through honeypots.

A. Deception Tools

The main question here is how the deception tools work and what their basic purpose is. The hackers in the past used the same old and traditional pattern to attack and get into the networks. That pattern was well known, and it was easy to detect and defeat them. The new technologies help the attackers to enter into the network using different patterns, which not only makes it difficult for the security implementers to detect it but also makes it difficult for the security frameworks to defeat it.

Deception tools have three major tasks:

- They examine the patterns of the attackers using different methods of investigation.
- They gather information about the attacking patterns.
- They counteract these attacks using the same pattern that the intruder used, which makes it easy to defeat them.

The machines use decoys, a type of firewall. The attackers are directed to a decoy system which reads the attacking pattern and prepares a response according to the pattern. The malware has to pass this decoy before entering the system [4]. The decoy is made using machine learning that teaches it how to act like the potential attacker and ensure the security of the system.

The deception network has duplicate servers and endpoints that work as a trap for the attackers. The attackers unknowingly enter into the mini-traps and decoy endpoints before entering the actual network. These decoy endpoints and servers collect information about potential malware. They then prepare a respective response that is effective in defeating that malware. The attackers get confused because the real assets and decoys look almost the same. The attackers become more cautious, which forces them to make a mistake. A single mistake makes all their efforts useless and defeats their attack. A number of different decoys are available that work differently according to the need. A network usually has a few hundred decoys in it to defend it against any cyberattack.

Cybersecurity has been more effective in using deception tools. There are two major deception tools that have been used by security implementers since the 90s: honeypots and honeynets. Honeypots are the traps or decoys used in the networks that present themselves as the potential target to the attackers. They usually pose as a high-value server or a real asset to the target. The cyberattacker unknowingly tries to get access to the apparent useful information in the honeypots. While they are busy with their attack, these honeypots gather information about the intruder's patterns and the ways they are obtaining information from the system [5]. The attacker's patterns are revealed, which makes it easy for the security designers to create security frameworks to defeat such attacks. The honeypots or decoys usually have the ability to detect the type of method used to get into the system and decide the response needed against that system.

Usually the honeypots are deployed multiple times in the network for the purpose of cybersecurity. If an attacker passes by the first honeypot, there is another one placed on the next level of the network where he might get trapped. If he manages to pass by it, too, there is an additional honeypot on the third level of the network. It is very difficult for the attacker to get into the real assets of the internal network while the honeypots are present. When the attacker accesses the honeypot, he strives to enter into it and send the malware while the honeypot strives to get information about the attacking pattern as much as possible. Once that is done, the honeypot releases the respective responsibility for the attack, which puts the efforts of the intruder in vain.

When an attacker tries to access the honeypot and fails, he realizes the tighter security must be implemented due to the importance of this system for the enterprise. This further urges him to invade the system using advanced and sophisticated attacking methods. Though he may succeed in accessing the dummy assets, his attacking patterns are also revealed. Most of the large and high-security organizations of the world use honeypots to mislead the attackers. The information gathered by these honeypots is used by the security researchers to design more secure and sophisticated security frameworks against the cyberattacks [6].

B. How a Honeypot Works

A honeypot is made to look like a real asset and usually contains a computer, data, and application. The system is isolated, no user is allowed to access the honeypot, and it is closely monitored. Any access to the honeypot is considered hostile. This is why each access to the system is closely assessed and studied to gather useful information about it. The attackers are busy logging activities while the system collects information about them. The whole cybersecurity mechanism distracts the attackers from the real assets. Cybercriminals also use honeypots to distract researchers, employing fake honeypots to study the behavior of the security researchers. This helps intruders employ different and more effective attacking strategies. The fraudulent honeypots directly work as decoys in the system and spread fake information, disturbing the research, which ultimately helps the cybercriminals in their purpose. The security researchers are now being more careful about the security mechanism and are now aware of different strategies being used by cybercriminals for different types of cybercrimes.

IV. Honeypots

There are two main types of honeypots: research honeypots and production honeypots. Research honeypots are the decoys that collect information about the attacking patterns and provide the information to the data analysts. The data analysts place unique data into the research honeypots which collect and analyze the data about the attackers and their patterns. There may be more than one attacker trying to invade a system at once, and research honeypots also try to find out the connections between these attackers. Research honeypots are simply made to collect information about the activities of the intruder as defined above, for research purposes. Although all the information collected by other honeypots is also used for research purposes, the research honeypot uses the information to help the researchers destroy the intruders [7]. Research honeypots can be implemented by law enforcement agencies to gather information about criminals and their activities. They are designed to keep the attacker engaged as much as possible; the more the interaction, the more the information is beneficial for research purposes.

Production honeypots are more important and effective honeypots placed inside the production network. They are implanted with the IDS. The data placed in the production honeypots are very similar to the real data but not useful for the attackers. While the attackers are occupied by the production honeypots, administrators study the attacking patterns and discover any type of vulnerabilities in the production system [8]. If a vulnerability is found, it is resolved and mitigated in real time to get rid of potential attackers and the threats associated with them.

There are other types of honeypots that are used in different security fields:

- Pure honeypots
- Low interaction honeypots
- High interaction honeypots
- Malware honeypots
- Email honeypots
- Spam honeypots
- Spider honeypots
- Database honeypots

A. Pure Honeypots

A pure honeypot is a complete copy of a production system. The honeypot usually contains mirroring data which shows the attacker that the honeypot has a great deal of sensitive data. The attacker believes that he has gained access to a real production system and will now be able to get useful information while in reality, he has fallen into a trap. For example, a pure honeypot is an instance of an investment and banking website where there are complete functions of a banking website available. The functions of this website are so real that it is nearly impossible for a novice attacker to notice he has been in a honeypot. The only difference between a production system and a pure honeypot is that the pure honeypot does not have the capability to connect to a real database and real customers. This honeypot collects much data about the attacker and wastes his time and resources [9].

A pure honeypot is usually much more useful than a research honeypot because it is more engaging. When an enterprise senses an attacker, it directs him toward the pure honeypot where each and every activity of the intruder is closely monitored. If the attacker is blocked, he suspects the enterprise security and never returns with the same attacking pattern. It is necessary to keep things running smoothly so that the attacker does not know that he has been monitored by the security professionals and researchers.

B. Low Interaction Honeypots

As the name suggests, a low interaction honeypot is designed to have less interaction with the attacker. These honeypots contain basic and frequently requested services by the attacker. This type of honeypot is usually easy to maintain and less risky. [10]. It is relatively inexpensive because it does not provide the functionalities as the production system. The basic purpose of a low interaction honeypot is to distract the attacker from the actual assets. These honeypots are scalable and are designed to change their capabilities and be upgraded. Low interaction honeypots are usually implemented in enterprises where security is not a very big problem. Most of the small-scale general domain organizations have low interaction honeypots where the attackers are also expected to be intermediately skilled [11].

C. High Interaction Honeypots

A high interaction honeypot is a copy of a production system with maximum interaction capability. The attacker is engaged for most of the time so that his resources and time can be wasted while important data are collected about his activities. These honeypots usually have slow response time so that the intruder's attack can be slowed down while making him believe that he is in an actual production system. These honeypots are used in organizations where research and development are the main priority. Organizations with medium-level security use high interaction honeypots.

D. Malware Honeypots

A malware honeypot is a specialized honeypot that collects malware. These honeypots usually simulate actual production resources that most of the malware tries to exploit. It simulates application program interfaces (APIs) of websites with loopholes and vulnerabilities. While the malware falls for these honeypots, they are being monitored and studied by the research team to find out how they attack and what they contain.

E. Email Honeypots

An email honeypot is used to catch the adversaries using emails. With this type of honeypot, an email is published that has the capability to detect spam emails and links. Such emails are published on the websites of enterprises. The attackers may get into this trap. Email scrapers that pick up the users who are not aware of their spam activities can be found and defeated using email honeypots.

F. Spam Honeypots

Spam honeypots show attackers the open resources which can be used against the enterprise as spam. The spammer uses these honeypots without knowing that they are caught by the monitoring team. Beginner attackers may fall for the trick, but the novice spammers are mostly aware of these mini-traps.

G. Spider Honeypots

Spider honeypots are basically made to trap the web crawlers that crawl through the websites and look for the invisible links. Many websites have invisible links that may contain a virus. Most of the users unknowingly click on these links and get trapped. Spider honeypots help to detect these invisible links. Once detected, the users can be intimated not to fall for this trick later.

H. Database Honeypots

Database honeypots are used to find the database attackers. A fake database is created with full functions of a normal database except the security features. The attacker thinks that the enterprise unknowingly left the database unsecured and he breaks into the database. All of the operations he performs produce fake results, but he doesn't know that he has been fooled by the security researchers. While he uses the database, his activities are counted and monitored to determine what he was actually trying to locate in the enterprise's database. [12].

V. Physical vs. Virtual Honeypots

Honeypots are very useful in the security of a network. Although they use a lot of resources, the advantages associated with them leave the cost of the resource behind. The main advantages of using honeypots as deception technology in the networks are:

- The honeypots collect real data. The data collected after an attack on the network may not be as effective as the data collected in real time. This includes the monitoring of all the services and operations the attackers performed. The sequence of the activities of the attackers can be studied and transformed into a pattern that can be used against the attacker later.
- Honeypots reduce false positives. The deception technologies usually generate false positive results. It includes the alerts of cybersecurity that may not be very useful for the administrator. The other deception tools usually generate many false positives that not only waste the resources but also waste the time of the security administrators. Honeypots potentially reduce the number of false positives and make the system more effective.
- Honeypots are cost-effective. The honeypots act like real assets, but they are not. As they do not have to maintain a high volume of data, they are cost-effective. All they must do is to look for the attacks and handle malicious activities. Honeypots are actually a good investment for the security of an enterprise.
- Honeypots capture encryption. Honeypots are programmed to act like a real asset, but they also can assess and maintain different types of attacks. They can even capture the encrypted data, which is usually not possible for most of the deception tools [13].
- Honeypots have disadvantages, too. Although the disadvantages do not reduce the actual benefits of honeypots, they still must be considered. Some of the disadvantages are:
- Honeypots collect data only when they are accessed and attacked by hackers. Zero attempt to attack does not help the honeypots collect any data. This means that the honeypots cannot predict an attack.
- Honeypots can only collect data when attacked. If the intruder suspects a honeypot in a network, he will never try to attack it, which means that the honeypot will not be able to detect and collect information about the attack.
- Honeypots are usually distinguishable from the actual production systems. The experienced hackers can easily differentiate between the honeypots and the legitimate production system, which means that the honeypots are not a completely effective trap for the attackers. They can track and discover the honeypot using the system's fingerprinting technique.

The need for honeypots cannot be denied. They have provided promising results in most of the security fields. Since honeypots have become a necessary part of the security of every enterprise, it is important to know what type of honeypot should be implemented. A wrong decision could cost the enterprise much more than the right decision. The basic idea includes a physical honeypot where a separate computer and application are deployed in the network. The sole purpose of the network is to wait for the attacker and record his activities during his interception. Using a separate computer as a honeypot requires an extra level of protection, maintenance, and monitoring all the time, which is more costly for enterprises.

With the increased level of competition in the market, every enterprise is now striving to achieve a competitive position while controlling its expenses and costs. Virtual honeypots can serve greatly in that purpose. A virtual honeypot is exactly the same as that of physical honeypot, with the exception of the base computer. A physical honeypot runs on a separate computer while a virtual honeypot runs on the same computer where the actual production system or the real asset is placed. A virtual honeypot does not require a separate computer and application. It is provided with a virtual place where the honeypot can perform its operations easily. The operating systems like Windows and Linux facilitate running a virtual environment on an actual system. The virtual environment saves the cost of a separate computer for the honeypot, which in turn saves a lot for the enterprise.

Virtual honeypots are definitely far superior than physical honeypots. First, the virtual honeypot costs almost nothing as compared to a physical honeypot. There is no need to buy another computer and implement another application suite. Second, implementation of an existing computer is always easy and more time-saving than buying a new one. A final reason virtual honeypots are a better choice is that the enterprise does not need to hire or employ another or an existing security researcher over a new honeypot. Besides these, there are a number of minor benefits that cost a lot when considered separately. Most of the enterprises today are using virtual honeypots for better and less costly security [14].

VI. Honeynets

Honeynets contain two or more honeypots placed at different locations in a network which are virtually or physically connected together. The basic idea of the honeynet evolved in 1999, when 30 of the popular security professionals in the world collaborated. In June 2000, the honeynet project started and became a nonprofit organization in 2001. The evolution of the honeynet is as follows:

- Detection Toolkit (1997)
- Single sacrificial computer (1999)
- Generation 1 Honeynet (2000)
- Generation 2 Honeynet (2003)
- Honeyd Software (2003)
- Distributor Honeynets (2004)
- Dionaea, Kippo, and Kojoney (2009)

The basic purpose of the honeynet project is to provide a platform for security professionals to research and find ways to make the honeynets and honeypots more effective and transparent. A honeynet is an architecture rather than a product. It is made of many actual computers, applications, and services for the purpose of confronting and defeating the attackers.

Honeynets include high interaction honeypots as decoys whose basic purpose is to engage the attackers as much as possible. While the adversaries are engaged, the honeynets collect as much information as they can about the activities and tasks performed by the attackers. The honeynets are isolated from the actual assets and no one among the users of the network access it, so anyone entering into the honeynet is a potential suspect for the security architecture. The honeynets are programmed to collect data about the activities of the attackers in real time, which can be used against the hackers later. Honeynets collect and analyze all the packets entering or leaving the network. These packets contain the information sent by the adversaries to access and collect information from the network. All the packets entering the honeynets are naturally suspected and captured [15].

A. Architecture Requirements of Honeynets

The two basic architectural requirements of honeynets are data control and data capture. In data control, the data flowing to and from the honeypots are controlled through the honeypots. The packets from the attackers are allowed to enter the network without any restrictions. The honeypots are more attractive to the attackers and they are urged to enter. Data capture works to seize the information from the packet. The packet enters the honeypot without any restriction and leaves the honeypot after being scrubbed by the honeypot. The connections of the packets by the adversaries are also made limited when the packets leave the honeypot.

B. Honeynet Generations

There are basically two honeynet generations, Generation 1 and Generation 2. Generation 1 was the initial generation of honeynets, not much evolved, simple, and had a limited capability. It was not able to handle a high level of traffic by the adversaries. The purpose of making it was to handle the automated attacks. As the attackers started invading systems with human-powered methods, the Generation 1 honeynets became ineffective. Generation 1 honeynets run on OSI layer 1 and use a reverse firewall for data control, which means that the data flowing out of the honeypots are confronted with the firewall to check further and analyze it. As this generation was a new and basic one, any skilled hacker could fingerprint it easily. This was the reason that Generation 1 Honeynets were abolished by security professionals.

Generation 2 is a newer generation of honeynets, which is more complex than the Generation 1 honeynets, using newer, more complicated technologies for data control and capture. This generation can examine the outbound data, which means that the data controller is far better than Generation 1. Generation 2 honeynets have the capability to decide whether to allow the data to pass uninterrupted or to block or modify the data [16]. Generation 2 honeynets run on OSI layer 2, which means that they are more effective than the Generation 1 honeynets. Most of the modern honeynet technologies used today are based on the Generation 2 honeynets created a decade ago.

Creating a honeynet is not difficult at all. All it needs is a system with an unpatched version of Red Hat Linux or Windows with an external internet. A station is created as a network monitoring system between the computer and the internet connection. This box silently records all the activities and traffic passing through it. Once this box is set up, the honeynet is ready. Now all the security professionals have to do is to sit back and wait for the attack to happen. It is necessary to continuously monitor the honeypot; otherwise, the risks can be inevitable. If a system is left unmonitored and vulnerable, the attackers may use this system to break into other systems. This type of attack is called downstream liability. A honeynet helps to reduce the downstream liability and eliminates the risks of an attack on other systems connected to a network.

An example of a honeynet is a backward firewall. This firewall does not restrict any traffic coming towards it. Instead, it restricts outbound connections. This protects the network from an attack, which is the basic purpose of a honeynet. Although honeynet restricts the outbound connection, it also alerts the attackers about a trap. When the outbound connection of an adversary is blocked, he suddenly gets alerted that he has entered a booby-trapped system. A clever hacker will instantly clean the disk and will never come back to that system again. This means that the outbound connection restriction feature of the honeynet makes it ineffective for security [17].

C. Implementation of Honeynets

Honeypots are used as a trap to catch the bad guys trying to access the network. A honeypot is easy to be implemented in a network, but it needs continuous monitoring. In order to implement a honeypot, the first step is to make a fully functional network. This network should include all the important aspects and necessary functions. A data management zone (DMZ) is made, which is a secure server that got its name from the demilitarized zone which is a zone in the army used against the enemies as a barrier [18]. This server includes a web server, FTP server, and DNS server. All the important assets and production systems of a network are placed in the DMZ, which acts as a buffer zone between the local network area and the insecure or less secure network in the enterprise. A DMZ provides the facility of secure services to different applications like web applications and FTP services. It is necessary to implement continuous monitoring and data capturing mechanisms because they will work to find out important information about the attack. A honeypot is generally useless because it does not perform any function. It just sits back and lets the bad guys do whatever they want while the security analysts monitor and conduct research on the attacking patterns in order to defeat them. A honeypot basically functions to distract the attacker from the actual production system and waste his time and resources into creating useless fake production systems and assets. DMZ is made fully secure because it contains an actual production system and assets. The security implementers try to leave no loophole in the DMZ, which can be used by an attacker later [19]. While the DMZ is made fully secure, the honeypots and the security layer around them is left open from many ports. The purpose of leaving the ports open is to distract the attacker and make him think that the security implementers have unintentionally left these ports blank. These open ports directly urge the invaders to attack the network which lands them into a loop of honeypots. Some of the security implementers leave all the ports of honeypots open which makes the attackers suspect that this is the honeypot. The ports like FTP, SMTP, or UDP are some of the very important ports of any network and can never be left open unintentionally. Therefore, a wise decision is to leave some of the non-important ports open so that the attackers do not know that this is a honeypot.

Whenever a honeypot is attacked, it clearly means that the network has been compromised. A honeypot is located in a network. When an attacker invades a network, he has come all the way through different security hurdles and has reached the point where he is very near to the actual production system. Once honeypots have been implemented in a network, it is necessary to conduct the security auditing to find out how secure the network is and what loopholes have been left behind consciously or unconsciously. For the purpose of the security audit of the network, Network Mapper (Nmap) can be used. Nmap is an open source utility tool used for the purpose of network auditing [20]. It discovers the vulnerabilities in a network and can also be used for network inventory management, service upgrading, service uptime, and host monitoring. It uses raw IP packets to find out which hosts are using networks and what services have been used by that host. It also defines what types of security features, like firewalls, have been used and what has been blocked by the filters.

The command of Nmap can be used in the command prompt with the command “nmap IP Address –sT”. Nmap command runs the functions of “nmap while the –sT” scans the TCP connections. The results of this command define who tried to access the port and which port was compromised. It shows attack time, date, the port of attacker, remote id, and the local port and IP that was attacked. The results can briefly explain what attracted the attacker to the network. Once found, these areas can be improved for better security.

D. Security of Honeynets

The security of honeynets should be considered, too. There should be a properly configured alert system for the honeynets. The system should generate creative and useful alerts that have very minimum false positives. The logs of all devices of the honeynets should be recorded onto the central log server so that it can control and monitor the honeynet. The security staff should be available all the time so that if anyone tries to get access into the system, he can be treated instantly. The immediate treatment in such case is close monitoring of the attacker’s activities so that his patterns can be recorded [21].

The honeypots should be made secure, but there should be some loopholes left behind intentionally. The attacker always attacks the assets with less security before proceeding to the most secure assets in a network.

E. Problems with Honeynets

As the backward firewall method has not been very effective in preventing the bad guys from making connections to other systems, the security professionals have made a new technology that can prevent the downstream liability. They have created an adaptive firewall rule which allows only five to ten connections each hour. It refreshes every hour. This limit can prevent the intruders from being suspicious because they can make one or two connections at a time which can help them to prevent being pointed out. This limit is also not very useful for the attackers because they do not get enough time to do serious damage to any other system. This limited time connection facility has considerably reduced the downstream liability while increasing the effectiveness of a honeynet. Each and every technique has its own positives and negatives.

The problem is that if the honeynet does not allow the attacker to make outbound connections, the security maintenance team will not be able to collect useful information about the attacker's patterns. If the honeynet allows the attackers to make an outbound connection, it helps to collect and analyze information about the attacking methods and techniques, but it also increases the risks for other systems. The data capture also faces challenges in a honeynet. Data capture collects information about the attack and saves it in the data log of the honeynet. This data log can be easily deleted by the attackers. This means that the attackers can delete the information collected about their patterns. This creates a problem for data capture techniques. The honeynet can also send the duplicate log file to the remote server of a network which will create a duplicate alert, and a duplicate alert is not given much priority to be checked.

Encryption has also proved to be a problem for security professionals. In the past, the attackers used to log into the system using simple text protocols which were very easy to be captured and recorded. The introduction of cryptography has also affected their way of attacking the systems. The bad guys are now using encryption techniques like SSH, which makes their communication to the network more hidden. This hidden communication is very difficult for the network to monitor, which means that the attacker can easily invade the system. The response to this problem is very interesting. The honeynets modify the operating system of the target computers. The keystrokes and transferred files are logged into another monitoring system which makes their encryption useless. The honeynets use steganographic techniques to avoid being detected by the targets. This resolves the encryption problem to much extent [22].

Honeynets solve many problems, and data reduction is one of them. A typical server has legitimate traffic where it is difficult to differentiate among the normal traffic and the attack traffic. The server gets confused, and the attacking traffic is usually passed without any issue. A solution proposed to this problem is the IDS, but even this system rarely works because many false alarms and alerts are generated which usually go untreated. Honeypots are the best solution to this problem. In a honeypot, there is no legitimate traffic, which means that any traffic passing through the honeypot is definitely an activity by an attacker. Honeypots can easily catch the bad traffic and stop the network from being compromised. It easily captures the data, which helps to find out what damage the attacker did or was planning to do.

The statistics of the honeynet project show that attacking incidents have increased many times in past years, which means that the attacking protocols are becoming more sophisticated and effective. The attackers now use point and shoot tools that contain pluggable exploits. This increases the chances of success of their attacks. It is necessary to use the honeypots on an IP address rather than the network's internal address. If a honeypot is placed on an IP address of a network parallel to the mail server of a company, better information about the attack can be captured [23]. The honeypot should have the same operating system, application suite, and patch level as that of the network assets that are being protected. If a honeypot is an exact copy of the real assets, better intelligence about the attacking patterns can be found. When a honeypot gets compromised, the security professionals can get to know about the loopholes in their actual production machine.

It is also very difficult to analyze the data gathered by the honeypots to find out what the attacker has done or what he intended to do. It is estimated that around 30 to 40 hours are required to analyze and find out what the adversary did in just 30 minutes. This means that the honeypots are not security appliances that can be just implemented and forgotten. They need regular maintenance and continuous testing to be more effective.

Honeypots are now being shifted from physical systems to virtual systems. In the past, honeypots were made of real systems that contained a complete operating system, application suite, and patch level. To decrease the cost, the honeypots are now being made on virtual systems where a single system allows more than one virtual computer to run on it individually. The systems like Linux and VMware allows this feature. A single host system can sponsor multiple virtual systems to run on it, which drastically reduces the cost of a honeypot. When a honeypot is made on a virtual system, it is also easy to make it a copy of a production system. What's more amazing is that the complexities of a physical honeypot system are reduced to much extent by using a virtual honeypot [24].

As the virtual honeypot runs on the same computer as the actual system, the disks are the files being created on the host system. Whenever a file or the honeypot gets compromised, the files are immediately wiped out to avoid any loss of important data. Virtual systems also support other functionalities like suspend and resume in which a compromised system freezes instantly to find out the attacker's processes. Smaller enterprises use virtual systems to make honeypots, while larger organizations do not rely on virtual systems. Instead, they create new physical systems for their network's security.

Honeynets also help the organizations to find out the employees destroying the security of it internally. If there are more than 50 or 100 employees working in an organization, it is possible that there will be two or three bad employees who may be helping the intruders to enter the enterprise's network. They may be weakening the internal security of the organization. Such hostile insiders may destroy the network and harm the organization. The best way to find out about such intruders is to increase monitoring. Strict monitoring can help to find out who is accessing something that he should not be. For example, the monitoring of the accounting system of the enterprise can help to find out who is trying to get into it with bad intentions.

VII. Honeynet Project

The Honeynet Project is a security research organization dedicated to investigating the field of security and making tools and techniques that can help to improve the security conditions of the networks. It is a non-profit organization working on an international scale. The organization has discovered thousands of new and advanced methods of attacks and malware created by the adversaries. It has also created advanced security tools to fight against the latest attacking techniques. The organization was founded in 1999, and the founders and contributors of the organization include expert security professionals from all over the world. It provides information about hacking attacks and malware across the world. The three main pillars of the Honeynet Project are research, awareness, and tools.

The research includes the shared efforts of collecting data about the attacking patterns on networks. This step includes the research portion where the areas like attacks, attacking motives, attacking patterns and communication during attacks are thoroughly studied to find out how the development in the area of hacking has been going and what is needed to fight against that development. The security researchers of the project from all over the world keenly look for the attacks on networks and define why the adversaries attacked the network and what they were trying to access. The results of this research are shown to the world using the blog posts of the project and the Enemy whitepapers. The organization also publishes a scan of the months where the latest attacks of the month are published.

Awareness is the process in which the world is made aware of the consents of attackers and the threats posed by them. The organization engages the security community of the world to define how they lag behind in the field of network security and how they can keep up with the latest developments in the security field. The public is also educated about the threats to their systems by hackers. The basic purpose of awareness is to let people know that can be the real target of attackers and that attackers continuously try to compromise their security. Through the information provided by the organization, people can understand how they can take basic measures to mitigate the ever-increasing risks and advanced threats that usually pass through the normal defense mechanism. The information is provided by the blog posts and Enemy series of papers.

The Honeynet Project also engages with the security community in security tool development via Google Summer of Code. The organization also assists other organizations wishing to conduct their own research in this field. They provide tools and techniques to these organization, which can help development in the field of security of networks. The tools of the Honeynet Project are available on the tools site of the organization. Recent development in tools include Capture-HPC, HoneyC, Cuckoo, Honeywall, and Honeyd. The development in tools is also published in the paper of the organization named "Know Your Tools."

In order to know more about how the Honeynet Project works, it is necessary to read some of their blog posts.

A. GSoC 2018 Infection Monkey Project

The Infection Monkey Project was presented by a student named Vakaris Zilius and mentored by Daniel Goldberg. According to their research, the Infection Monkey is a security tool that is used to find out how much a data center is resilient to the infections of internal server and perimeter breaches. It is an open source tool where anyone can contribute to it. The monkey in the Infection Monkey self-propagates throughout the data center using different methods and presents the results of its propagation to a centralized server called Monkey Island server. The new research in the project added a new framework in the tool that is used to find out the web vulnerabilities. The researchers also added new exploitation methods to the Infection Monkey tool [25]. These new developments empowered the Infection Money in networks and increased its functionalities.

The main features added to the Infection Monkey tools are:

- Remote code execution in the HADOOP server with default setting and YARN.
- A remote exploit for Oracle WebLogic API.
- A remote exploit for Struts2 Jakarta multiparser RCE exploit.
- SSH key stealing that allows the Monkey to move in the network using the SSH keys for authentication.
- A new framework that all these exploits use to easily develop new web attacks.

Vakaris, the researcher fixed different issues and found different vulnerabilities. The main issues found were:

- It's still hard to add new exploits, as it requires adding code to multiple ancillary files like the report generation mechanism.
- Python 2.7 is going to die, and new useful libraries are Python 3 only, so we'll need to migrate the code.

The main challenges faced by the researchers are:

- The codebase was most challenging. It was difficult for the researchers to research topics like Python, network communications, and NoSQL database.
- Setting up the exploitable network was the most challenging task for the researchers.

- The implementation of Web RCE exploits was also challenging for the researchers. It required refactoring, which was a difficult phase.
- The vulnerability of WebLogic was difficult to implement.
- The setting and implementation of HADOOP was also a harder task for the researchers.

The learning for the researcher in the field are:

- Code quality.
- Python introduction and learning.
- Web server debugging knowledge.
- Networks basics.
- Improvement in the knowledge of git.
- Knowledge about web security.

B. GSoC 2018 Project: Conpot

This project was completed by Abhinav Saxena. The main achievements of this research project include:

- Implementation of FTP (RFC 959) and TFTP (RFC 1350) protocol stacks based on gevent.
- Refactoring of an existing telnet library to be compatible with the Conpot codebase.
- Implementation of an abstract file system that proxies and wraps an actual file system by providing os.* wrappers.
- Bug fixes and refactoring of the existing BACnet and IPMI protocol stacks.
- Migration of codebase from Python 2.7 to Python 3.5.
- Wrote an internal interface implementation that introduces a decorator, allowing protocol servers to interact more deeply with each other.
- Helping users with issues and pull request reviews.
- Bug fixes in auxiliary Docker files.
- Wrote 123 unit tests and refactored all existing 44 unit tests, increasing coverage from 44% to 72% at the time of this writing [26].

The future work scope defined in the project is as follows:

- The Telnet server implementation will be the next task after getting a Telnet library compatible with the Conpot.
- In the future, there can be a central authentication system which will be common and will bring more consistency to the system.
- The Generic database will be created to support the logging issue.
- The serial server implementation will be easy in the future as the python code has already been generated that is compatible with this system.
- The type of hints support will also be made in the future.

The main challenges that the researchers faced are:

- The migration of Conpot from Python 2.7 to Python 3.5 was challenging for the researchers.
- The understanding of the internal interface has also been challenging for the researchers.
- The implementation of a comprehensive file system was also difficult.
- It was difficult to find a library compatible with RFCs reading.
- The FTP server implementation with Python was specifically challenging for the researchers.
- Running the Conpot with non-sudo privileges was difficult.
- Bug challenges and hunting was difficult.
- Reverse engineering packets by the Wireshark and reconstructing it was difficult for the researchers.

The learning outcomes achieved by the researchers for the project were:

- Python code skills.
- The learned difference between Python 2 and Python 3.
- End product delivery.
- RCFs knowledge.
- File system implementation.

The two blog posts show how a researcher conducts research and presents results in the form of a project summary. There are thousands of other researchers that have presented in the form of blog posts.

C. Dionaea Honeypot

This research has been conducted by Tan Kean Siong on Dionaea honeypot. Dionaea honeypot is an old honeypot that supports SMB. The honeypot is an open source system and supports many security features but lacks the support for latest WannaCry Malware. This malware has not been detected by most of the old security defence mechanisms. The SAMBA RCE is also not caught with the Dionaea honeypot. The work done in this research has made the honeypot able to catch these types of attacks. Dionaea is a server-side honeypot based on low interaction. It captures the copy of the malware. The honeypot was created in the era when the Conficker worm was most deadly. The strong SMB network of the honeypot enabled it to handle many other new and latest malware [27].

WannaCry is a ransomware that was released in 2017 and affected millions of computers across the world. It specifically targeted many organizations. Most of the computers with the Windows operating system were attacked by this ransomware. This malware first released the EternalBlue exploit and then installed the DoublePulsar as a backdoor tool. The malware was expandable and could target other computers using the internet of the first target. The technique used to handle this ransomware in this research is to face it. The Windows PC with DoublePulsar backdoor is used where the incoming command of DoublePulsar is dissected. The worm releases its payload to the honeypot which contains the WannaCry ransomware.

The outbreak of SambaCry occurred seven years ago. It was discovered after WannaCry ransomware. This vulnerability got the attention of the whole world as it had the capability to attack the Linux systems, IoT devices, and NAS. This research also explained how Dionaea could defeat SambaCry. The SMB Open AndX requests are accepted by the Dionaea Honeypot. The honeypot accepts the payload from this ransomware and collects it to be studied later.

D. Legal Considerations

Just like any other field, deception technologies have their own legal complications. There are potential wiretapping laws that affect deception technology. Generally, there is no documented case law, but this technology can be blocked under the banner of consent laws. For example, there should be a description of the system that states that whoever uses this system can be wiretapped and monitored without any prior notice. This type of notice can help security implementers to avoid any legal battle against them.

VIII. Difference between Honeypots and Honeynets

Honeypots and honeynets are basically the same. A honeypot is a single decoy in a network designed to deceive the adversary while the honeynet is the set of many honeypots placed at different places in a network to deceive the attacker. Both of them have the same purpose. The only difference is that a single honeypot can be less effective than the honeynets in capturing the attackers. Both honeypots and honeynets are implemented in larger networks as parts of network IDSs. Even if there is no banner like this on the systems, there is still no ethical consideration for the adversaries if they try to access the systems of an enterprise without their permission.

IX. Deception vs. Cooperative and Competitive Technologies

There are a number of traditional cybersecurity mechanisms that are helping organizations. Technologies like end-point encryption and firewalls help to avoid unwanted malware, but these systems generate many alerts, and generally, most of these alerts are useless. The number of alerts sometimes increase so much that it reaches millions.

It is naturally impossible for the security implementers to look and take care of each and every alert. Among these millions of alerts, if a single alert is potentially harmful, it can destroy the whole network and make it ineffective. The alerts produced by the deception technologies are the result of a binary process. The honeypots create alert only when any unwanted traffic requests to pass through it. If the alert is created, it means that there is a risk to the production system. The results of the deception technologies can be either 0% or 100%. These mini-traps can be stated as the area secured with the laser. Whenever anyone interrupts the laser, an alert is created, which should ensure that there has been a try to ping and enter the area without any prior consent.

Deception technologies are not standalone technologies. The tools in the deception technologies work along with other cyberdefense technologies. The traditional cyberdefense technologies provide security at a general level while the deception technologies provide security at advanced levels [28]. Deception technologies strengthen the existing cyber defense technologies and help to secure it in depth. A standalone deception tool may not be very effective. Different partners in the cyberdefense add another protection layer which makes the security mechanism more sophisticated, helps to defeat adversaries, and helps secure the large networks of the enterprises.

X. Intrusion Detection System and Deception Technologies

The traditional security tools and mechanisms help combat general malicious activities. The normal advanced threats are treated by the IDS, while the sophisticated and complicated threats and malicious activities are generally treated with the deception technologies. The IDS is the system that closely monitors the traffic running through the network and looks for suspicious activities. If it finds any suspicious activity, it generates an alert for the security implementers on the basis of which they can take some actions against it [29]. These systems can block the traffic coming from a suspicious IP address.

The problem with the IDS is the false positives it generates. The IDS causes many alarms while most of the alarms among them are useless. The false alarms can waste the time of security implementers. The organizations need to continuously maintain and update the IDS so that it can keep fighting with malicious activities. The IDS generates false alarms because an alarm is deployed whenever malicious traffic is suspected. The suspicion of the IDS is mostly wrong because most of today's attackers have become more sophisticated. Deception technologies are far better IDSs in terms of false positives [30].

The IDS is made to find any suspicious traffic and report it. It is not programmed to defeat or defend against potentially harmful malware. Deception technologies are programmed to not only monitor the traffic but also to take respective actions against the activities of attackers. The results of the operations of the deception tools are put into the binary operations which reduce the number of false positives. Deception technologies reduce the necessity of the IDS. It can also sometimes enhance the capabilities of the IDS.

XI. Conclusion

Deception technologies have played a vital role in network security. The technology has increased the development speed of the security mechanism of networks and brought it parallel to the technologies being used against the networks. Honeypots and honeynets are used as research tools where the information captured is used to find out how to keep up with the attacking patterns of different intruders. Deception technologies have improved the security mechanisms and contributed to enhancing the capabilities of traditional security frameworks like IDSs. With the old security mechanisms, the defenders had a 99% chance to defend themselves while the attacker had only a 1% chance to attack. The deception technology has changed it, and now the attacker has a 99% chance to attack while the defender has only a 1% chance to defeat it. The Honeynet Project has been playing a very important role in research in the field of network security. The organization has contributed to major research in the field.

References

[1] R. C. Joshi and A. Sardana, Honeypots: A New Paradigm to Information Security. Boca Raton, FL: CRC Press, 2011.

[2] J. M. Marín, J. A. Naranjo, and L. G. Casado, "Honeypots and honeynets: Analysis and case study," in Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance. Hershey, PA: Information Science Reference, 2015, pp. 452–482.

[3] N. C. Rowe, "Deception in defense of computer systems from cyber attack," in Cyber Warfare and Cyber Terrorism, L. J. Janczewski and A. M. Colarik, Eds. Hershey, PA: Information Science Reference, 2007, pp. 97–104.

[4] B. Endicott-Popovsky, J. Narvaez, C. Seifert, D. A. Frincke, L. R. O'Neil, and C. U. Aval, "Use of Deception to Improve Client Honeypot Detection of Drive-by-download Attacks," in International Conference on Foundations of Augmented Cognition. Berlin, Heidelberg: Springer, 2009, pp. 138–147.

[5] S. Garfinkel, "All about honeypots and honeynets," 2003, May 01, Retrieved from: https://www.csoonline.com/article/2115901/all-about-honeypots-and-honeynets.html

[6] M. Rouse, "Intrusion Detection System (IDS)," 2018, Retrieved from https://searchsecurity.techtarget.com/definition/intrusion-detection-system

[7] M. Dargin, "Increase your network security: Deploy a honeypot," 2017, Retrieved from Network World: https://www.networkworld.com/article/3234692/increase-your-network-security-deploy-a-honeypot.html

[8] M. Rouse, "honeypot (computing)," 2018, Retrieved from: https://searchsecurity.techtarget.com/definition/honey-pot

[9] J. Spacey, "11 examples of a honeypot," 2017, Retrieved from: https://simplicable.com/new/honeypot

[10] E. Amoroso, "An introduction to deception technology," 2018, Retrieved from https://www.helpnetsecurity.com/2018/12/06/introduction-deception-technology/

[11] K. Sadasivam, B. Samudrala, and T. A. Yang, "Design of Network Security Projects Using Honeypots," Journal of Computing Sciences in Colleges, vol. 20, Jan. 2005, pp. 282–293.

[12] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, Computer Security: Principles and Practice, Upper Saddle River, NJ: Pearson Education, 2012.

[13] W. Stallings, "Cryptography and Network Security: Principles and Practice. Upper Saddle River, NJ: Pearson, 2017.

[14] N. M. Chowdhury and R. Boutaba, "Network Virtualization: State of the Art and Research Challenges," IEEE Communications Magazine, vol. 47, Jul. 2009, pp. 20–26, doi: 10.1109/MCOM.2009.5183468.

[15] W. A. Conklin, G. White, C. Cothren, R. Davis, and D. Williams, Principles of Computer Security (4th ed.). New York: McGraw-Hill Education Group, 2015.

[16] G. B. White, E. A. Fisch, and U. W. Pooch, Computer System and Network Security. London: CRC Press, 2017.

[17] A. Singhal and X. Ou, "Security Risk Analysis of Enterprise Networks Using Probabilistic Attack Graphs," in L. Wang and S. Jajodia, Network Security Metrics. Cham: Springer International Publishing, 2017, pp. 53–73.

[18] M. Akiyama, M. Iwamura, Y. Kawakoya, K. Aoki, and M. Itoh, "Design and Implementation of High Interaction Client Honeypot for Drive-by-Download Attacks," IEICE Transactions on Communications, vol. 93-B, May 2010, pp. 1131–1139, doi: 10.1587/transcom.E93.B.1131.

[19] M. Kim, M. Kim, and Y. Mun, "Design and Implementation of the Honeypot System with Focusing on the Session Redirection," in International Conference on Computational Science and its Applications. Berlin, Heidelberg: Springer, 2004, pp. 262–269.

[20] C. K. Dimitriadis, "Improving Mobile Core Network Security with Honeynets," IEEE Security and Privacy Magazine, vol. 5, Aug. 2007, pp. 40–47, doi: 10.1109/MSP.2007.85.

[21] T. Sochor and M. Zuzcak, "Study of Internet Threats and Attack Methods Using Honeypots and Honeynets," in International Conference on Computer Networks. Cham: Springer International Publishing, 2014, pp. 118–127.

[22] I. C. Lin and T. C. Liao, "A Survey of Blockchain Security Issues and Challenges," International Journal of Network Security, vol. 19, Sept. 2017, pp. 653–659, doi: 10.6633/IJNS.201709.19(5).01.

[23] S. Rathore, P. K. Sharma, V. Loia, Y. S. Jeong, and J. H. Park, "Social Network Security: Issues, Challenges, Threats, and Solutions," Information Sciences, vol. 421, Aug. 2017, pp. 43–69, doi: 10.1016/j.ins.2017.08.063.

[24] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An Overview of Fog Computing and its Security Issues," Concurrency and Computation: Practice and Experience, vol. 28, Apr. 2015, pp. 2991–3005, doi: 10.1002/cpe.3485.

[25] Daniel.haslinger, "GSoC 2018 project summary: Infection Monkey," 2019, Feb. 05, Retrieved from Honeynet: https://www.honeynet.org/node/1372

[26] Daniel.haslinger, "GSoC 2018 project summary: Conpot," 2018, Aug. 18, Retrieved from: https://www.honeynet.org/node/1371

[27] Roberto.tanara, "Dionaea honeypot: From Conficker to WannaCry + SambaCry CVE 2017-7494, 2017, May 30, Retrieved from: https://www.honeynet.org/node/1353

[28] A. Kott, C. Wang, and R. F. Erbacher, Eds., "Cyber Defense and Situational Awareness," Advances in Information Security, vol. 62, Cham: Springer International Publishing, 2014, doi: 10.1007/978-3-319-11391-3.

[29] N. Lord, "What is threat detection and response? Solutions, benefits, and more," 2018, Retrieved from: https://digitalguardian.com/blog/what-threat-detection-and-response-solutions-benefits-and-more

[30] R. A. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, and Y. L. He, "Fuzziness Based Semi-supervised Learning Approach for Intrusion Detection System," Information Sciences, vol. 378, Feb. 2017, pp. 484–497, doi: 10.1016/j.ins.2016.04.019.