Bind Shell logic to bypass security controls

Done by : Majid AL Maashari
Email : omanras@gmail.com

Introduction:

The next generation security controls and solutions become very straight and have advanced intelligence to identify and detect abnormal behaviors which can harm the environment and gain control of the system. This white paper will illustrate some techniques to gain access through breakned system by using python bind shell as an example to pass system commands and bypass signature based and behavior based techniques.

Scenario:

The attacker exploits file upload vulnerability which allows the user to upload and execute any file extension in the linux environment running apache as HTTP service and php as web app. On other hand, the server has configured and limited php functions executions by closing any function allowed to pass system commands. Alsoe, the server has other security controls like anti-virus and next generation firewall and IDS solution.Moreover, the attacker does not have a public IP address which allows him to use reverse connection technique.

Gain access issues:

- **PHP function limitation:**
  Some server owners are closing or limiting the execution of dangerous functions such as (system, eval, symlink) which make the PHP web shell useless. Because of that, the attacker used other web shells programmed with python or perl also Ruby shell if it is installed on the server.

  But we can ask a question here. Why not use a C/C++ shell or even bash script ?. That is because it is easy to find or programme useful web shells with perl or python. However, that is not the main reason, the C/C++ system command function in some linux kernels required root privilege or user privilege to execute. Also the anti-virus by default alerts any executed function run from C/C++ programs because it is run under system level not on a sandbox environment like python. Also, it is hard to limit or close functions in python or perl and not hard to break any function limitation in python especially in version 2.7. In advance, it is better to not use bash script because it can be alerted from several security layers like antivirus and IDS and also from SIEM solution and it will be usually alerted under MITRE ATT&CK rules.

  NOTE: It is helpful to check and review the scenarios and attack behaviors in MITRE ATT&CK web site before script or program gaine access app. It can help you to design your app logic to bypass updated security rules based on behaviors if the target uses next generation security solutions.

- **Antivirus:**
  One of the most annoying security layers which needs advance knowledge about the antivirus logic which is not an easy task to know which type and version that installed in the target device.

  But why is it important to know which antivirus installed in the targeted device? That is because of the different logic and features between antivirus products which need specific techniques to bypass the security detection mechanism.However, the latest AV have the same main features and almost same detection mechanism.

  There are two common types of AV mechanism, first signature based and second behaviour based. Signature based logic can detect the attacker file based on file hash. The AV producers have a strong and updated list of malicious file signatures from different sources. That is why it is a very bad idea to use web shells from public sources.

  The attackers commonly bypass signature based by changing the content of the file by adding lines or deleting null lines from the code and there are other techniques by using encoding or encryption algorithms which are recommended by professionals. And it is recommended because the next generation security solutions used advanced algorithms with AI to build a strong signatures list. For example, some AI logic tries to divide the virus file and create a signature for each line at the same time generate another signatures for the lines by adding the common techniques which used to bypass signature based security.

  According to that, it is better to code your own shell and keep in mind there are other risks of using a public web shell, the attackers who are classified under cyber espionage or who have search engine dorks of the public web shells can find and use your file.

  On the other hand, the behaviour based mechanism uses a sandbox and tries to execute the file to check if the logic of the file matches with the suspicious behaviour rules. The professionals attackers often update their shells according to the latest news and behaviour rules which are published in the common and official security resources like CERT organizations or private organizations such as hacker news web sites or MITRE ATT&CK.

- **Firewall & IDS:**
  The next generation firewall and IDS has the ability to detect and analyse all layers in the OSI model. Also, there are some firewalls that support annoying features which are a hadec for the attackers. For example, there is a decrypt feature which converts TLS and SSL or other encryption protocols like IRC, even base64 and URL encode to plain text and analyze it.

  So the attackers try to bypass this type of security layer by using their own TLS or SSL encryption key certificates or IRC channels, but still the firewall and IDS detect the attacker traffic and deny it. Moreover, the next generation security solutions work like an orchestra so if the firewall detects unauthorized encryption traffic first the event will be denied second the IDS will analyse and block the devices or the port number which send the traffic and send an alert to the antivirus to block the application which sends the traffic.

  In this case the attacker needs a smart way to bypass the firewall by using an encryption algorithm which can't be detected.

Reverse shell VS Bind shell:

Before starting to explain in detail it is important for the attacker to choose which type of connection he needs to gain access and bypass the security controls. So, what is the difference between reverse shell and bind shell? As sample as that, it is the logic, in reverse shell the attacker receives the session from the targeted device which gives the attacker the ability to bypass the security controls, because it is normal for the server to connect with another device if it is under the firewall rules.

For example, if the attacker has VPS or any public IP address with FTP or HTTPS service he can easily bypass the firewall and pass system commands under covert channel because it is normal for the OS to open random ports and send outbound traffic. However, the attacker has to first bypass the AV second he has to offer an authentic way to pass the traffic for example using verified SSL certificate for attacker IP address.

It is hard to offer a public IP address which is not signed with a footprint leading to the attackers. Because of that, the professional hackers usually use hacked servers as proxies or zombies.

NOTE: The attackers usually use netcat / nc for reverse connection which is a bad idea if the target uses next generation security solutions. Netcat is one of the most alerted applications by the security controls.

On the other hand, Bind shell logic starts by opening a port in the target device then the attacker connects through that opened port. This process can be detected by security controls, so the attacker needs proper hardening steps to follow before executing the shell.

Generally the attackers gather information about the targeted device such as service, open ports, security controls and others like CDN servers or proxy servers like cloudflare which is haide the real IP address of the targeted device.

When the attacker is usually facing a target that's using cloudflare service he tries to identify the real IP address of the targeted device. In our scenario, the attacker is available to upload and execute any file extensions by exploit file upload vulnerability.In this case one of the common techniques is to use small php script to send a request to another web service host owned by the attacker, usually the attackers used free hosting web site service to reserved the request and identify the real target IP address.

However, often it is not significant to know the real IP address because you can reach your shell by domain name.But, that requires other information like which ports open in the real server and which ports open in the proxy server which is forwarded to the opened port in the targeted device which make using bind shell useless and easily detected by security solutions.So, it is more reasonable to know the real target IP address. If the attacker already got the real IP address of the targeted device he can go to the next step by opening a port in the target system which is needed to be careful with. If the attacker try to open port number like 4444 I can make sure that will be alerted by security controls under Metasploit observed, and if the attacker try to open port which is commonly known as default services number like 8080,3306 or 1433 it can be alerted as new service run. So, it is better to choose a random port number which is not usually used by any default services or other malwares.

NOTE: To choose the proper port number it depends on the security mechanism used by security controls. It is not important to follow the previous explanation as a reference. It is better to know which security control logic and rules are used in the target environment. In fact using bind shell technique in unknown security controls is like a gambling and the safest way to use a direct shell installed in a public area. For example using a web page shell for web page service used by the target.

Coding bind shell:
After gathering enough information about the target and security controls you can start to design your code logic and script your own shell, for the scenario which mentioned in the is white paper you can download the script from the link (https://github.com/omanras/bind-shell-logic).

The script content three files programmed with python 2.7. The first file is (genrate.py) which gives two options to choose (bind shell) or (linker shell).

- **Bind shell:**
  The logic of this part will generate a python file and ask you to enter file name and also port number and random number, the random number has two purposes: to encrypt the data and to change some variable names.

  NOTE:The logic need random number because of, the algorithm encrypted data start to convert the data to hex than convert hex to decimal in this step the script will add the random number value to the decimal to change the actual data, it is sample but effect encryption methodology, because it is hard to guessing the random number you cohesion and have no limits for the random number length.However, after that, the decimal number will be converted to base64 encode because the web proxy firewall allowed to pass base64 content.Another thing is when the firewall decode the base64 will got normal decimal numbers which can be normal content if it is hex data it can be alerted.Moreover, The random number used also to change the variables content to bypass signatures based mechanism.

  So, after generating the shell file and executing, you can connect to the uploaded shell through (cl-shell.py) which asks the server IP address, port number and random number which is configured in the uploaded file shell.

  NOTE:The generated bind shell file which is uploaded to the target system can be executed through both 2.7 or 3 python versions.

- **Linker shell:**
  The linker shell logic will generate multiple files. The first file will be a php file and the second python file and both of these files need to be uploaded and executed in the targeted server. The (generate.py) will ask you to enter three values (php file name, python file name and random number).

  However, it more stealthy to use linker shell because, the linker shell logic not require to bind port in the targeted device instead of that the shell will used open service like php on HTTP server and if you ask about the python file it is to bypass the php function limitation as mentioned in the scenario. In details, the linker logic shell will create two files in the targeted device (commands.txt, results.txt). In the (commands.txt) the php script will insert the data which is sent through (cl-linker.py). After that, the python script will read the data from (commands.txt) and execute it and save the results in the (rseults.txt) file. Than the php script will include the content from the results.txt in to the same php script body under HTML tag (<!-- -- >). Finally the (cl-linker) will collect the encrypted results from the php script and decrypt.

  NOTE:The encryption logic is the same as the bind shell algorithm.

  Remember: Do not use variable names like (password,key,username,..etc) because some security solutions will alert if the script file contains some sensitive keywords. That is why the bind shell algorithm used keyword (random) instead of key or password. Also, the recommendation is to not use an external lib or function which is required to be installed in the targeted device. The bind shell script is designed for learning and proof of concept, so please don't use it as an official solution.

Conclusion:

In conclusion, it is not easy but not hard to bypass the next generation security solutions and the challenge in growth between the red and blue. It is important to follow up with the latest cyber security news and latest updates in the information security field.