A study on Backdoor File threats over Web Applications

[1]Asish Joseph James, [2]Seena Augustine, [3]Ms. Lisha Varghese

[1,2] P G Scholars, Amal Jyothi College of Engineering, Kanjirappally, 686518

[3] Assistant Professor of Amal Jyothi College of Engineering

[1]asishjosephjames@mca.ajce.in, [2]seenaaugustine@mca.ajce.in, [3]lishavarghese@amaljyothi.ac.in

***Abstract* -- In this digital era we have to rely on web applications for our day to day life. So it is necessary to study and understand about various threats over the web applications in order to secure them from various kinds of attacks. Threats caused by backdoor files is one of the most disastrous kinds of attacks on web applications. In computing, a backdoor refers to any method by which authorized and unauthorized users are able to get around normal security measures and gain high level user access. Hackers can use a backdoor to install all manner of malware on your computer achieved by taking advantage of vulnerable components in a web application. Once malware is installed, detection is difficult as files tend to be highly complicated. Through this study we try to learn and understand about the effects of backdoor files on web applications in order to take necessary precautions to protect our web applications.**

Keywords—Backdoor, Backdoor threats, Weevely, PHP web shell

I. INTRODUCTION

Web application security is one of the several aspects of information security. It focuses on securing web apps and services from malicious actors who can take advantage of code errors, scripts, and other vulnerabilities to take control of the app or extract data. A backdoor file can be a serious threat to a web application. Backdoor file is a malware type that negates normal authentication procedures to access a system. As a result, the perpetrators gain remote access to resources within an application, such as databases and file servers, giving them the ability to remotely issue system commands and update malware. Installation of backdoor file is done by taking advantage of vulnerable components in a web application. Once installed, detection of this kind of files is difficult as files tend to be highly obfuscated.

In 2018, backdoors were the fourth most common threat detected for both consumers and businesses respective increases of 34 and 173 percent over the previous year. Built-in or proprietary backdoors are put in place by the hardware and software makers themselves as artifacts of the software creation process.Software developers create these backdoor accounts so they can quickly move in and out and These backdoors aren't supposed to ship with the final software released to the public, but sometimes they do. There are various methods for creating backdoor applications, Kali Linux tools like Weevely can be used to generate backdoor fi

II. LITERATURE REVIEW

Rajesh M. Lomte1 , Prof. S. A. Bhura [1] says about the different types of Web application attacks and possible preventive measures that can be taken in order to protect web applications from intruders.

Rina Elizabeth Lopez de Jimenez [2] focuses on the knowledge of the technique Pentesting on web applications, discusses the different phases. Most commonly these attacks can be victims as well as upgrades software tools to make a penetration test- or Pentesting and it also helps to detect vulnerabilities that are present in the web application and detect the active threats present in the web application.

Gurdeep Singh and Jaswinder Singh [3] did an evaluation on penetration testing tools which helped in understanding the effective Kali Linux tool that detects the code that cause harm to the web application.

Ossama B. AlKhurafi and Mohammad A. AlAhmad [4] possible backdoor vulnerability attacks on the web applications are examined and its effects are evaluated.

Truong Dinh Tu , Cheng Guang , Guo Xiaojun and Pan Wubin [5] conducted an experiment for finding and detecting webshell inside web application source code that are crucial to secure websites. Proposed a novel method based on the optimal threshold values to identify files that contain malicious codes from the web applications.

III. BACKDOOR

A backdoor is a type of malware that access a system by negating the normal authentication procedures and gain remote access to resources within an application, such as databases and file servers. Which give perpetrators the ability to remotely issue system commands and update malware. Backdoor installation is often achieved by taking advantage of vulnerable components in a web applications.

Once installed, detection of this kind of files are difficult as files tend to be highly obfuscated. Web server backdoors are used for a number of malicious activities like:

- Data theft.
- Website defacing.
- Server hijacking.
- The launching of distributed denial of service (DDoS) attacks.
- Infecting website visitors (watering hole attacks).

A. PHP web shells

Advanced persistent threat (APT) assaults PHP web shells Web shells are the scripts which are coded in many languages like PHP, Python, ASP, Perl and so on which further use as backdoor for illegitimate access in any server by uploading it on a web server. The attacker can then directly perform the read write and edit operation once the backdoor is uploaded to a destination, you can edit any file of delete the server file. Today we are going to explore all kinds of PHP web shells what-so-ever are available in Kali Linux and so on. So, let's get started. Kali Linux has inbuilt PHP Scripts for utilizing them as a backdoor to assist Pen-testing work. They are stored inside **/usr/share/webshells/php** and a pen-tester can directory make use of them without wasting time in writing PHP code for the malicious script.

- simple backdoor.php
- qsd-php backdoor web shell
- php-reverse-shell.php

B. Backdoor vs Exploit

While backdoors and exploits seem awfully similar at first glance, they are not the same thing. Exploits are accidental software vulnerabilities used to gain access to your computer and, potentially, deploy some sort of malware. To put it another way, exploits are just software bugs that researchers or cybercriminals have found a way to take advantage of. Backdoors, on the other hand, are deliberately put in place by manufacturers or cybercriminals to get into and out of a system at will.

IV. METHODOLOGY

According to the experiment conducted using Weevely (Kali Linux tool written in python), which is a command line web shell which is dynamically extended over the network at runtime, designed for remote server administration and penetration testing. Its terminal executes arbitrary remote code through the PHP agent with a small footprint that sits on the HTTP server. Over 30 modules shape an adaptable web administration and post exploitation backdoor for access maintenance, privilege escalation, and network lateral movement, even in the restricted environment.

A. Features

• Stealth tiny web shell.

• Post penetration tool.

• Helps to maintain access with the host after successfully compromising the host.

• Helps to edit and manage files in the host server.

B. Technique Employed

Weevely php stealth web shell and backdoor communications are hidden every communication between the server and client are hidden in HTTP Cookies. Communications between client and server are obfuscated to bypass NIDS signature detection. Weevely PHP stealth backdoor and web shell has more than 30 modules available for post exploitation tasks. Implements reverse shell connection from the host.

C. Implementation

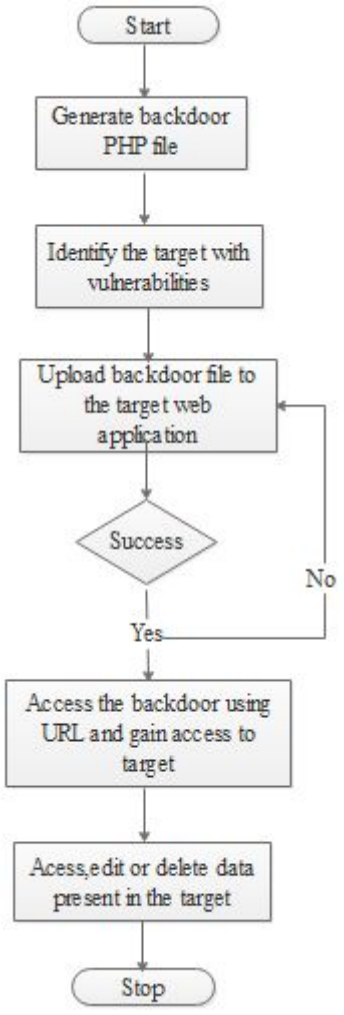

Fig: Flowchart to represent working of weevely

Step 1: Generate PHP shell using weevely. Generate shell with password and file name.

Step 2: Find target website. Find the target website with file upload vulnerability.

Step 3: Select file to upload. Select the shell file generated by weevely.

Step 4:Upload the shell. Upload the PHP backdoor file into the website

Step 5: Get access to the server from the kali terminal. Access the website server and files using the password protected backdoor file. List the files in the current directory using ls command.

Step 6: Move through directories in the server using cd command.The shell file is located in the uploads directory, move to the home directory using cd .. command.

Step 7: Access files in the server. Now you can edit, update, add or delete any file in the website server using basic Linux commands. Open index. php file using gedit.The index.php file is now opened in the geditor, any changes made in the editor will reflect in the server.You can make any kind of changes in the website now and get access to the database files, get all the data from them.

V. SECURITY MEASURES

Change your default passwords. If you leave that default password in place, you've unwittingly created a backdoor. Change it as soon as possible and enable multi-factor authentication (MFA) while you're at it. Keeping track of a unique password for every application can be daunting. A Malware bytes Labs report on data privacy found that 29 percent of respondents used the same password across numerous apps and devices.

Monitor network activity. Anyweird data spikes could mean someone is using a backdoor on your system. To stop this, use firewalls to track inbound and outbound activity from the various applications installed on your computer.

Choose applications and plugins carefully. Cybercriminals like to hide backdoors inside of seemingly benign free apps and plugins. The best defence here is to make sure all apps and plugins you choose come from a reputable source.

Use a good cyber security solution. Any good anti-malware solution should be able to stop cybercriminals from deploying the Trojans and rootkits used to open up those pesky backdoors.

VI. CONCLUSION

Overall the research identifies how the Backdoor is exploited using threats by penetrators to get into and out of the Web application. It is difficult to identify and protect yourself against built-in backdoors. More often than not, the manufacturers don't even know the backdoor is there. As technology develops, the exposure to attacks increases. The more aware the users are, the lesser vulnerabilities harm them. Hence it is important to use the technology wisely and the possible security features should be included in the Web applications.

VII. REFERENCES

[1] Survey of different Web Application Attacks & Its Preventive measures http://www.iosrjournals.org/iosr-jce/papers/Vol14-issue5/D01454651.pdf?id=7477

[2] Pentesting on Web Applications using Ethical Hacking https://ieeexplore.ieee.org/document/7942364

[3] Evaluation of Penetration Testing Tools of KALI LINUX http://www.academicscience.co.in/admin/resources/project/paper/f201609201474344177.pdf

[4] Survey of Web Application Vulnerability Attacks https://ieeexplore.ieee.org/document/7478735

[5] Webshell Detection Techniques in Web Applications https://ieeexplore.ieee.org/document/6963152

[6] Malvertising: A case study based on analysis of possible solutions https://ieeexplore.ieee.org/document/8365356

[7] HTTPS Hacking Protection https://ieeexplore.ieee.org/document/4221121

[8] Low-cost Detection of Backdoor Malware https://ieeexplore.ieee.org/document/8356377

[9] Survey of Hacking Techniques and it's Prevention https://ieeexplore.ieee.org/document/8392053

[10] True2F: Backdoor-resistant authentication tokens https://ieeexplore.ieee.org/document/8835225

[11] Webshell Detection Techniques in Web Applications https://ieeexplore.ieee.org/document/6963152

[12] Webshell Traffic Detection with Character Level Features Based on Deep Learning https://ieeexplore.ieee.org/document/8540857

[13] Lexical Analysis for the Webshell Attacks https://ieeexplore.ieee.org/document/7545259

[14] Training a multi-criteria decision system and application to the detection of PHP webshells https://ieeexplore.ieee.org/document/8842705