Shibboleth walkthrough

Index

List of pictures

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:

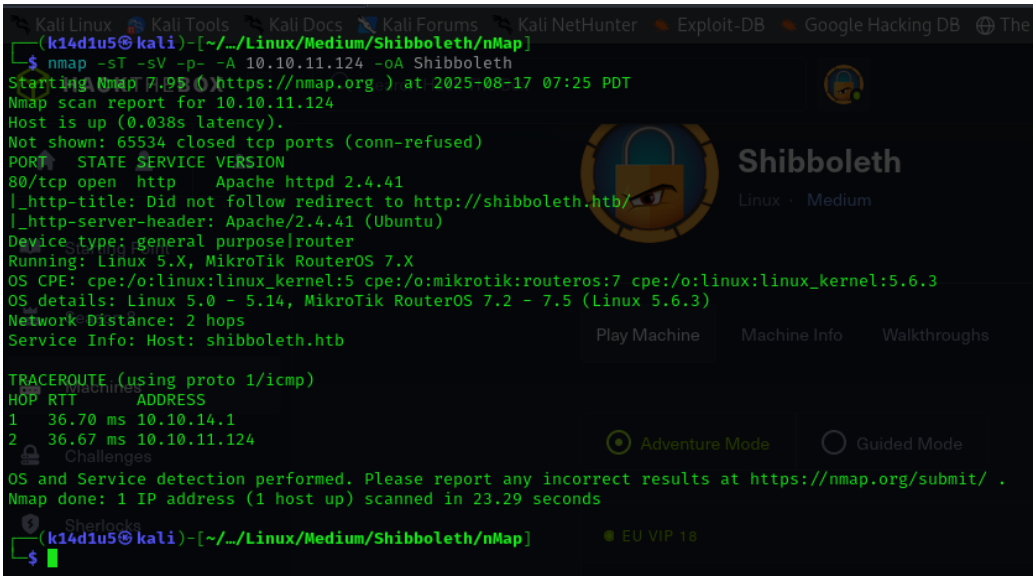

Figure 1 - nMap TCP scan results

TCP open port is 80. So, I just found a web application running on that port. Also, nMap recognize Linux as OS. For this box, it was useful the nMap UDP scan as well, which results are shown in the following:

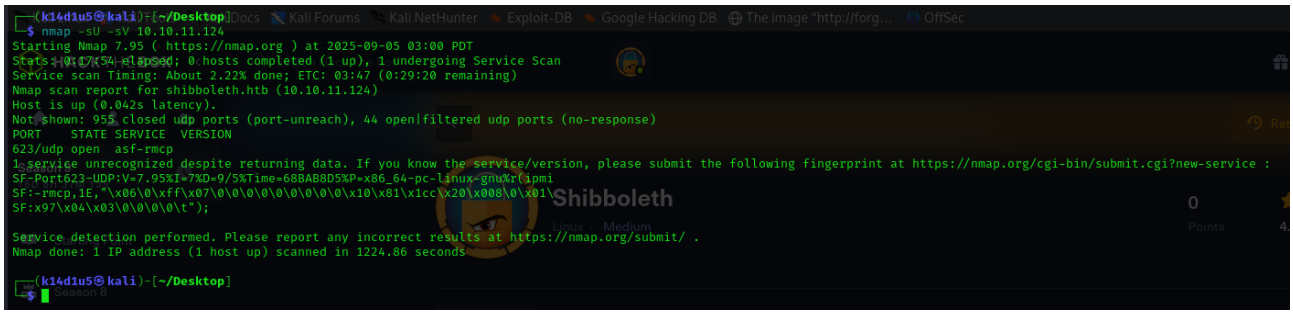

Figure 2 - Map UDP scan results

This scan showed that port 623 was open. nMap recognized it as $asf - rmcp$ service. I run again this scan only on that port and running nMap scripts (I forgot to take this evidence) and I found out that this service seems to be vulnerable to "IPMI 2.0 RAKP Cypher Zero Authentication Bypass" attack.

Initial foothold

Since the target could be exploited via IPMI, I tried to interact with him using $ipmitool$. I was able to establish a connection and enumerate channels, but I didn't find anything useful in this way. After a while I

tried to find something, I made the decision to check if Metasploit was able to support me for this task. Luckily, Metasploit provide a useful module to retrieve password hash via IPMI. Thank to it, I was able to obtain a password hash, as shown in the following picture:

Figure 3 - Administrator hash

Since I obtained a hash, I tried to crack it using *hashcat* tool and the *rockyou* wordlist. I was able to crack the hash and find the password in clear text:

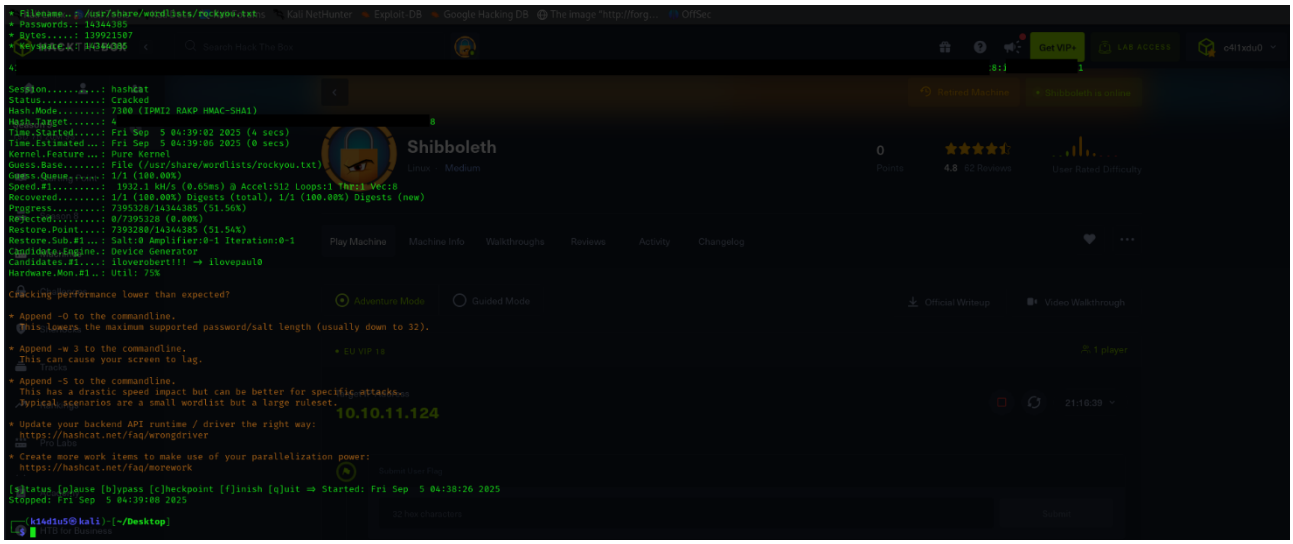

Figure 4 - Password cracked

Also, I run a task to find new subdomains on the web application and I found out three of them:

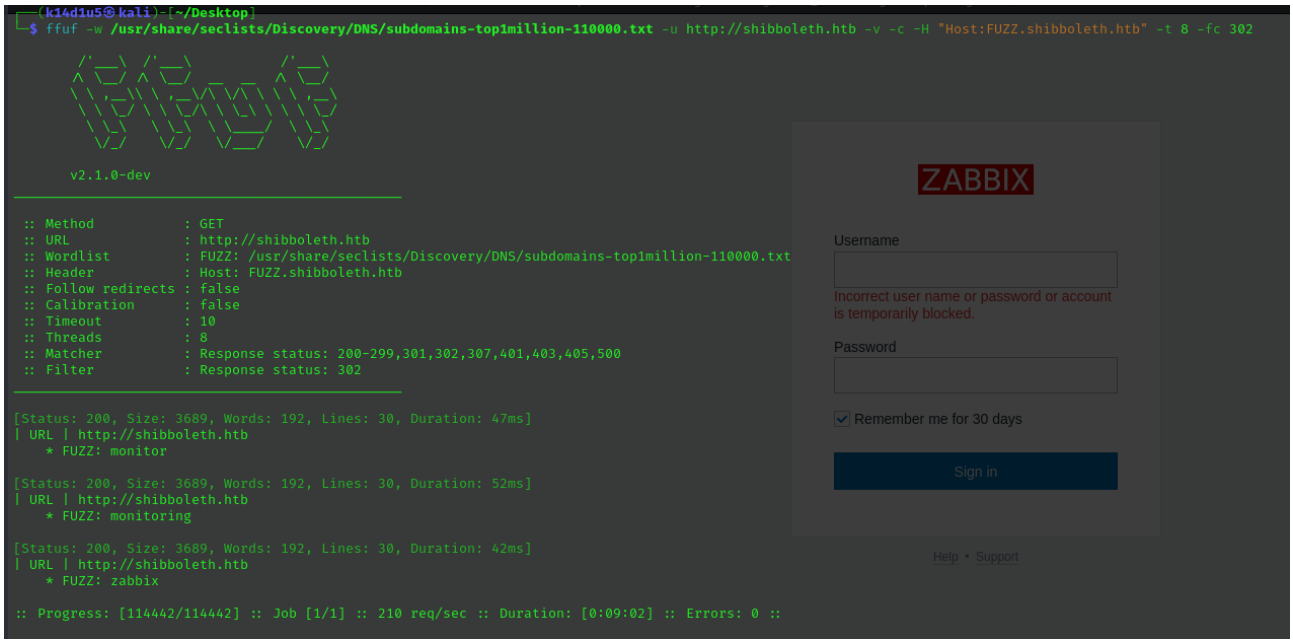

Figure 5 – Discovered subdomains

When I tried to access to them, I was browsed on a login page.

User flag

At this point I know which application is running, Zabbix, I had a login page and a pair of credentials. I don't remember if I tried to use the credentials in the login form, but probably I did and it was successful. Also, I looked for a known Zabbix exploit on the Internet and I found an interesting one. In fact, when I run it, I was able to obtain a reverse shell and I checked the $/etc/passwd$ file:

Figure 6 - Zabbix exploitation

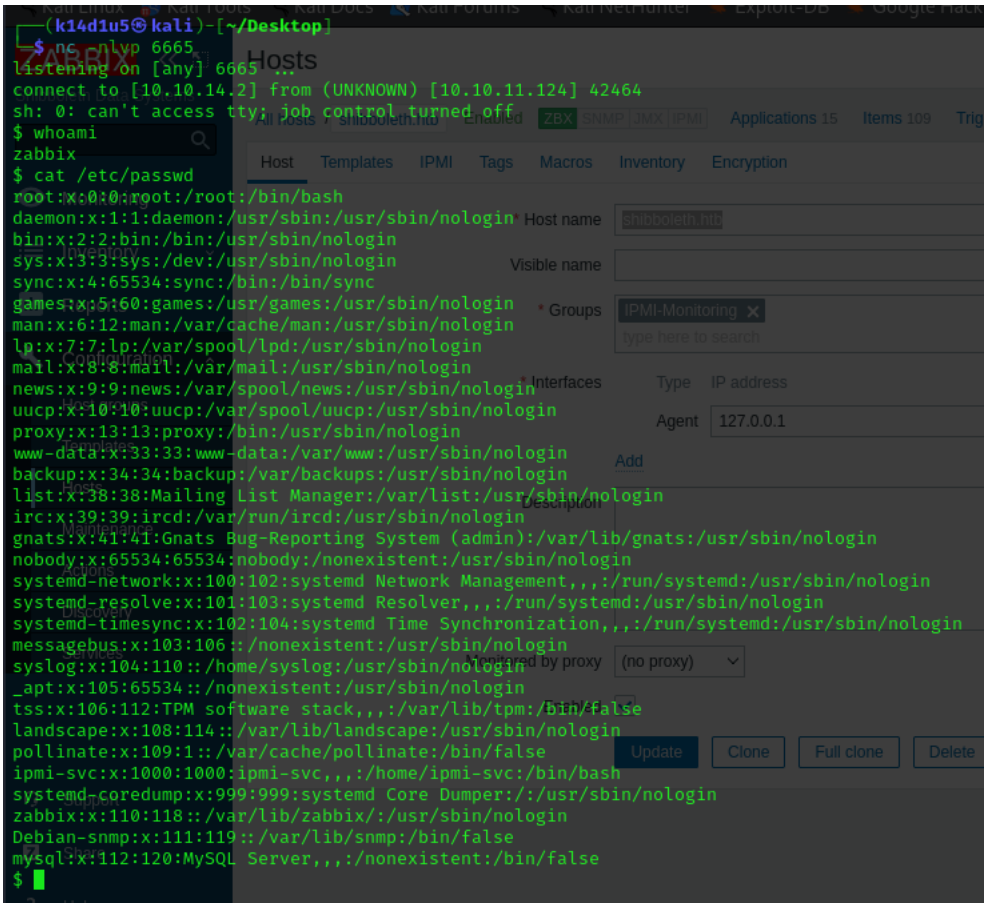

Figure 7 - Reverse shell as zabbix user and /etc/passwd file

I found out only one user, $ipmi - svc$. At this point, I tried to became that user via su command. To do so, I used the same password I already used for the exploit and it worked. In this way I retrieved the user flag:

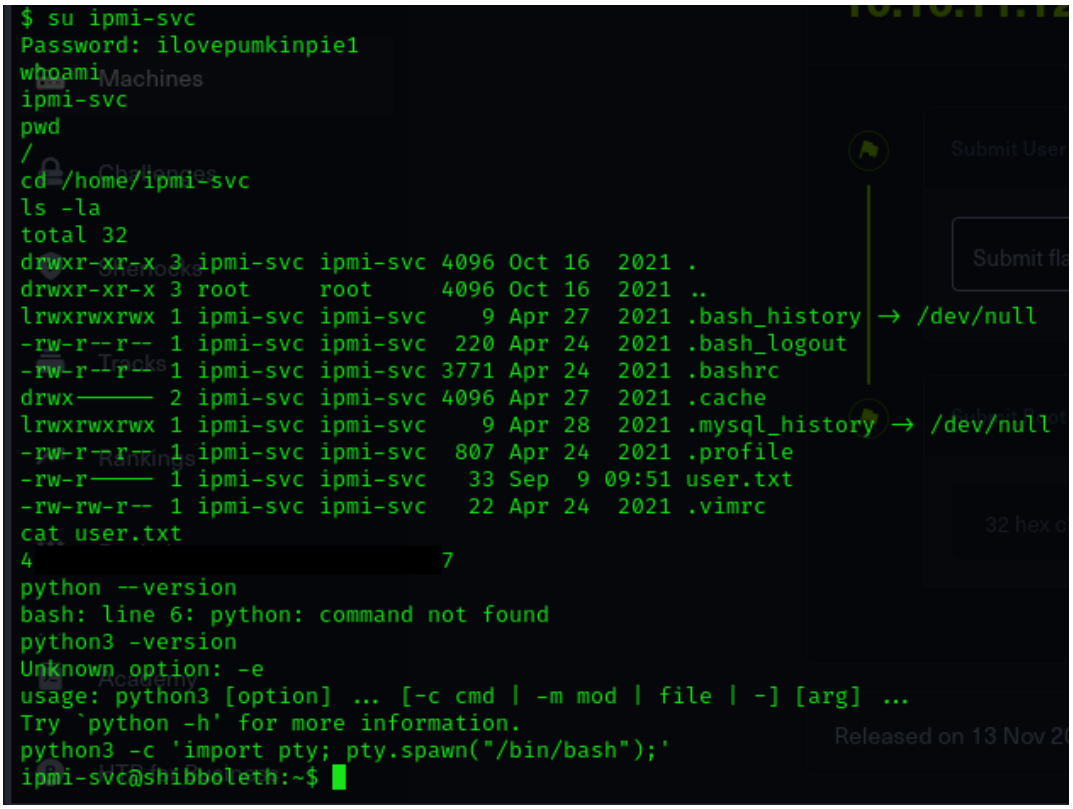

Figure 8 - User flag

Privilege escalation

I looked for some interesting file and information on the target. Honestly, I wasn't able to find something in a manual way and after a while I made the decision to run LinPeas tool. Its output was very interesting and it retrieved database credentials:

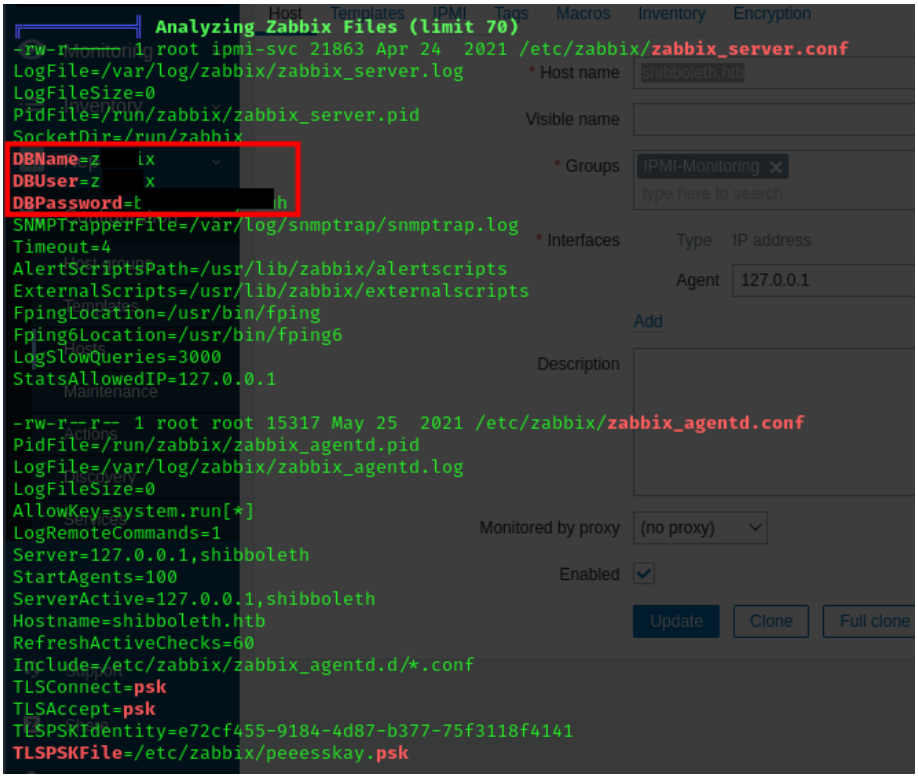

Figure 9 - Database information

At this point I connected to the database and checked which permissions I had:

Figure 10 - Database user permissions

Based on the permission I had granted, I looked for some exploit on the Internet. I found the CVE-2021-27928 and its exploit. So, I created via MSFVenom a payload to use:

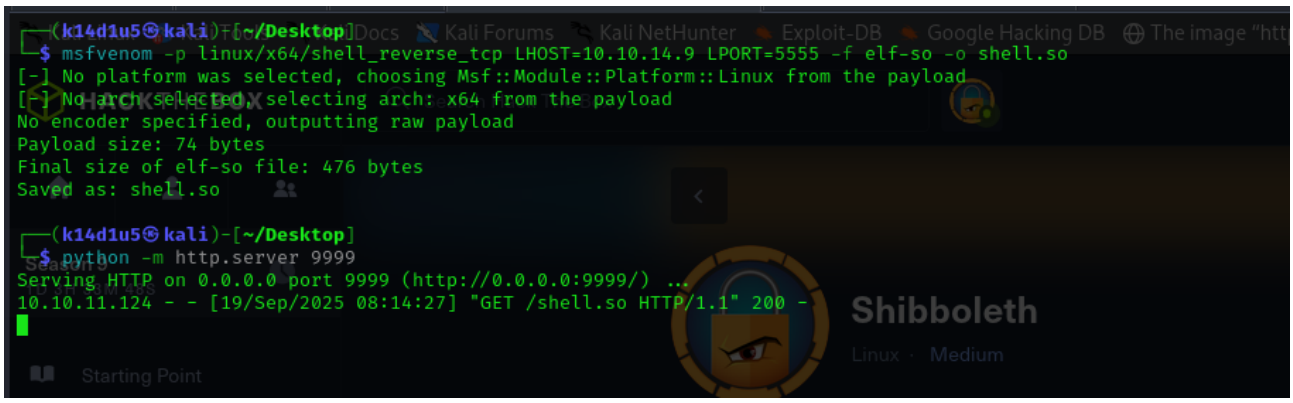

Figure 11 - MSFVenom paylod generation

At this point I uploaded it on the target:

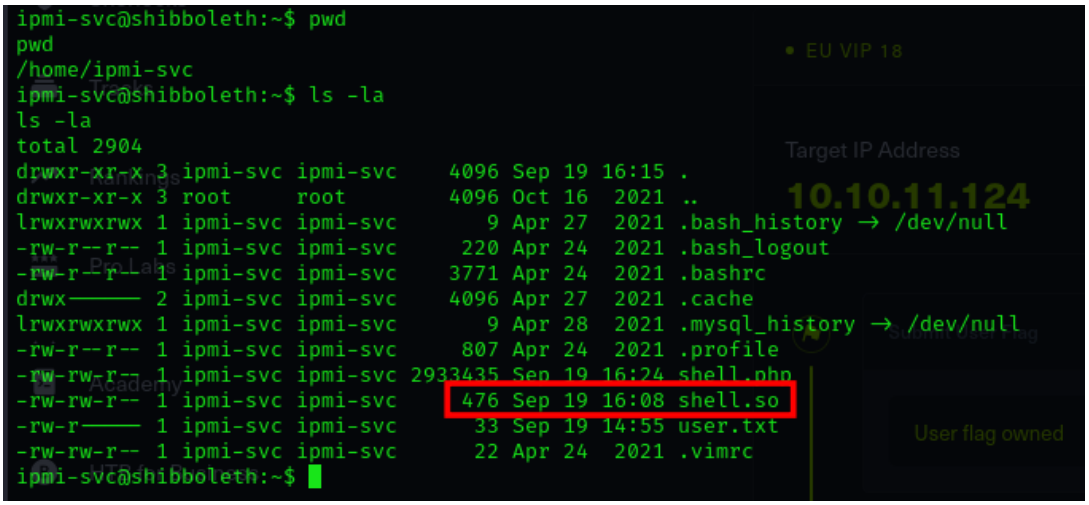

Figure 12 - Payload uploaded on the target

At this point I just needed to execute the malicious payload via SQL and retrieve the root flag:

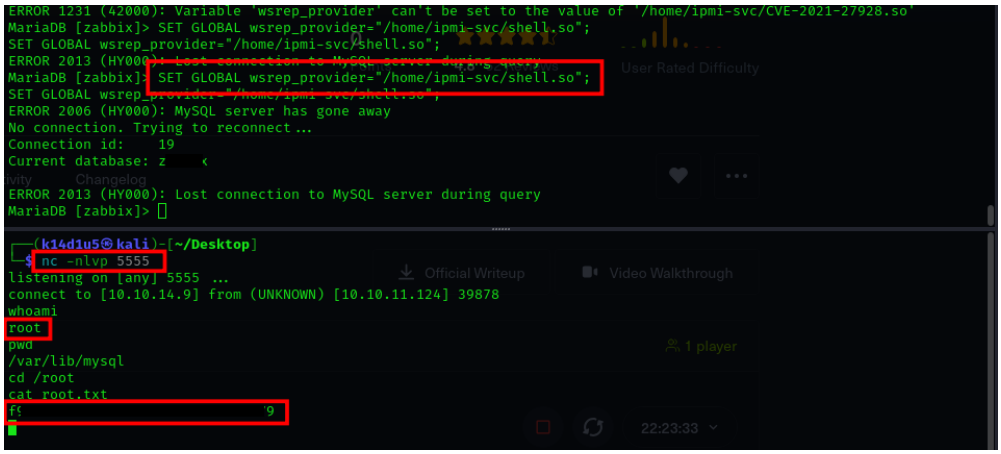

Figure 13 - Privilege escalation and root flag

Personal comments

This box was very interesting. I learn a lot about IPMI exploitation for the user flag. It was a completely new topic for me and I improved my skills, of course. Also, privilege escalation using SQL via User Defined Functions is always something to practice and keep in mind. I am very happy to complete this box that gave me more confidence and improved my performance. It is a very good box.

Appendix A – CVE-2021-27928

The CVE-2021-27928 affects a MariaDB unknown functionality. The manipulation leads to untrusted search path. It is possible to initiate the attack remotely. The manipulation with an unknown input leads to a untrusted search path vulnerability. Using CWE to declare the problem leads to CWE-426. The product searches for critical resources using an externally-supplied search path that can point to resources that are not under the product's direct control. Impacted is confidentiality, integrity, and availability.

References

1. CVE-2021-27928: https://www.cve.org/CVERecord?id=CVE-2021-27928;
2. IPMI testing guide: https://www.rapid7.com/blog/post/2013/07/02/a-penetration-testers-guide-to-ipmi/;
3. Zabbix exploit: https://www.exploit-db.com/exploits/50816;
4. SQL User Defined Functions: https://juggernaut-sec.com/mysql-user-defined-functions/;
5. CVE-2021-27928 exploit: https://github.com/Al1ex/CVE-2021-27928.