Table of Contents

EXECUTIVE SUMMARY

This document details the security assessment (external penetration testing) of corporate environment with 5 servers. The purpose of the assessment was to provide a review of the security posture of corporate environment with 5 servers, as well, as to identify potential weaknesses in its infrastructure. Overall, we noted that basic information security principles were not adhered to. For example, complex passwords, and input validation were not in use. As well, many of the vulnerabilities we discovered were aided by outdated software. Patch management and keeping software up to date is a key area to improve on. To facilitate these changes, we recommend thorough review of NIST 800-14 ' 'Generally Accepted Principles and Practices for Securing Information Technology Systems' and the OWASP '2017 Top 10' is strongly recommended (links to both resources included in the References).

DETAILED FINDINGS

Challenge 1 - Mark's mail

- Find the flag on marks mailbox @ mail.ctf
- Reconnaissance phase
    - Mailserver IP is 10.10.10.6 (nmap)
    - ports 993 and 143 open (nmap)
    - username : mark (SSH login attempt revealed this)
- Exploitation phase
    - Used hydra to perform a dictionary attack on mail.ctf, port 993, username mark (obatined earlier earlier), password turtle (obtained via dictionary attack)
    - Ssh in to 10.10.10.6, ls to view flag, cat to open flag.txt

Challenge 2 - Search.ctf (300 points)

- http://search.ctf
- Reconaissance phase
    - Search.ctf IP is 10.10.10.8 (nmap)
    - Port 80, 22 open
    - Found username root via ssh to 10.10.10.8
    - Ran dirb to see what I can find out (enumerate) about search.ctf directory structure
        - Using a dir text file, scanned server for open directories (code 200 configured it existed), discovered /search/ folder, and was able to reach hidden search box field (not in source code of) search box..
- Exploitation Phase
    - Dictionary attack on the the web server password on port 80 (http-get), username root -- discovered password 123456
    - Attempted to Ssh root@10.10.10.8 , password 123456 **rejected
        - Still trying to find a way in via root credentials acquired..
    - Possible approach to own target: use LFI/RFI to make the search display contents
        - Created simple php script, wrote http://search.ctf via curl, now trying to get our php script to execute.. **unsuccessful

Challenge 3 - Best Coffee website

- Portfolio, Kevin's Website, Best Coffee
    - Find the Flag @ http://coffee.ctf/site/index

    - suggested approach - use curl, LFI vs RFI

- Reconnaissance Phase
    - Nmap to disc. Open ports 80, 22,
    - Coffee.ctf site IP is 10.10.10.10
    - 22/tcp open, ssh using OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)

Challenge 4 - Portfolio.ctf exercise http://portfolio.ctf

Reconnaissance phase:

- Use curl to modify the database
- Hints..."Database"-map, Contact_me.php
    - Use sql injections, use sqlmap to dump, --dump
- Basic info:10.10.10.7, Apache/2.4.18 (Ubuntu),
- Possible vulnerabilities to exploit (discovered via nmap and hydra)
    - + OSVDB-3268: /img/: Directory indexing found, + OSVDB-3092: /img/: This might be interesting…
- Ran dictionary attack on 10.10.10.7 - found lots of usernames/passwords for root, kevin… none work (tried soccer, kevin etc) from SSH

Exploitation phase:

- Was able to find the contact_me.js file through dirb - modify this to own.. http://10.10.10.7/js/contact_me.js

SUMMARY AND RECOMMENDATIONS

Challenge 1 – Mark's Mail, mail.ctf

- We were able to crack Mark's password in under 10 minutes, using a simple dictionary attack. We recommend that Mark adopt more complex passwords (e.g. passwords containing letters, numbers, symbols, and avoiding words in the dictionary), particularly for users like himself with root level access to the mail server. Doing so would greatly increase the work rate required to crack his password via a dictionary or brute force attack, dissuading an attacker from attempting such an attack.
- Another recommendation for users with root level access is to implement multi factor authentication. This way, even if an attacker got a hold of Mark's password, they would need a secondary piece of information to log in. An example would be a hardware token, such as a Yubikey, that would generate a one time password to supplement his normal password to login to the mail server (See References).
- Mark should consider anonymizing his SSH login screen. When we attempted to log in via SSH, we were greeted with mark@xxxxx . We surmised that this was his true username, and it allowed us to cut down our time to breach the mail server in half.

Challenge 2 – Search.ctf

- Unlike Challenge 1, the username presented upon logging in via SSH was obscured. The SSH shell displayed a username of` 'root'. We tried to pursue this lead by performing a dictionary attack to crack root's password – and were successful. But the username of root and passwords obtained did not grant SSH access.
- We were able to enumerate the directory structure of the site to reach the hidden 'search' box, embedded in the website. Hidden elements such as these should not exist, as they present a latent

threat that could be exploited by an attacker. If a site element is being modified or is under development, it should not be included in the production version of the site.

- While we were not able to capture the flag on search.ctf, it was clear that through Remote File Inclusion (RFI, a type of injection attack), we would eventually be able to access the webvservers directory and files. Given more time, the exploitation would have proceeded as follows. To perform the RFI attack, we would upload a php script (designed to permit file access) to the web server via the curl tool. Once our code was uploaded on the site, we would manipulate the URL to call our script instead of the normal Search output – a text file.
- To prevent these kinds of attacks, we recommend some sort of input validation, or fuzzer, to prevent injection attacks like these.

Challenge 3 – Kevin's Site

- We performed some initial reconnaissance, which revealed that OpenSSH 7.2.p2 was in use. Our research indicates there is an easy to execute exploit available for this version of OpenSSH – username enumeration. Knowing all the usernames that existed on web server would have made a dictionary attack on the root password much easier (Exploit Database, 2016). Out of date software, such as OpenSSH and Apache, should be patched immediately. Patch management strategies should be reviewed as well.

Challenge 4 – Coffee site

- Our limited time with this challenge did not permit us to fully exploit this site. However, we were able to locate the source code for the contact me form (contact_me.js) through directory enumeration. This file was not protected, and did not require authentication to access or view. Basic site authentication for important site elements is a 'must'.
- Using a dictionary attack on the usernames 'kevin' and 'root', we were able to obtain valid passwords for both accounts. Using these credentials, we likely could have modified the contact_me.js file through Local File Inclusion (LFI). LFI is appropriate in this case since the file already resides on the server, and we had no issue obtaining user credentials.

Final Recommendations

- Conduct vulnerability assessment at least twice a year and penetration testing
- at least once a year or if there is a major change in the information assets.
- Develop and implement a training path for the current IT staff.

Conclusion

The results and 'lessons learned' during this capture the flag exercise demonstrated the importance of knowing the basics. Simple information security principles, like enforcing a complex password requirement for all accounts (particularly root ones), were not adhered to. As mentioned earlier, this makes for some very 'low hanging fruit' for attackers, and should be addressed immediately. Several resources should be reviewed (see, for exmaple, NIST 800-14 'Generally Accepted Principles and Practices for Securing Information Technology Systems' (Swanson and Guttman 1996)). While our time with the environment was limited, would be attackers are not limited in the same way, and can take their time performing reconnaissance, and working their way up the kill chain. Our failures should not be taken as successful defence – they merely reflect the constraints of the exercise.

REFERENCES

"0_o." "OpenSSH 7.2p2 - Username Enumeration." *Exploit Database*, 2 July 2016, https://www.exploit-db.com/exploits/40136/.

OWASP. *OWASP Top 10 - 2017*. OWASP, 2017, https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf.

SANS. "Writing a Penetration Testing Report." *SANS InfoSec Reading Room*, 1 Apr. 2010, https://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343.

Swanson, Marianne, and Barbara Guttman. *Generally Accepted Principles and Practices for Securing Information Technology Systems*. NIST SP 800-14, National Institute of Standards and Technology, 1996. *CrossRef*, doi:10.6028/NIST.SP.800-14.

Yubikey. *Unphishable Secure Multi-Factor Authentication*. 2018, https://www.yubico.com/why-yubico/for-business/.