Australian Government
Australian Cyber Security Centre

ACSC
AUSTRALIAN CYBER SECURITY CENTRE
PROTECT

July 2015

Securing Content Management Systems (CMS)

Introduction

1. The security of external-facing infrastructure is critical for organisations when considering the security of their network as a whole. Even if external-facing infrastructure does not host sensitive information, there is still a significant risk to the reputation of organisations if external-facing infrastructure is tampered with.

2. Vulnerabilities within content management systems (CMS) installed on the web servers of organisations are often exploited by adversaries. Once the CMS has been compromised, the web server can be used as infrastructure to facilitate targeted intrusion attempts.

Intended Audience

3. This document outlines strategies for identifying and minimising the potential risk to websites using CMS software. The intended audience is individuals responsible for developing and securing websites using CMS.

Risks to content management systems

4. Adversaries can use automated tools to scan the Internet for web application vulnerabilities. If a vulnerability is found, an adversary can attempt to exploit it to gain access to the system. Typically these compromises are opportunistic and the result of the poor security posture of the victim rather than a targeted cyber intrusion.

5. Once a CMS has been compromised, adversaries can exploit their access to:

    a. obtain access to authenticated and privileged areas of the site.

    b. upload malware to the web server to facilitate remote access, for example web shells[1] or remote administration tools (RATs).

    c. inject malicious content into legitimate web pages. This could be used to serve exploits or malware to visitors or to facilitate remote access to the infrastructure.

6. Although the web server may only host publicly-available information, the compromise of any organisation's web server is significant as an adversary can exploit the trust of users of that website. An adversary can use a compromised web server as part of a 'watering hole' attack, or as command and control infrastructure to facilitate other intrusions, for example, compromising

[1] A web shell is a type of remote administration tool (RAT) that is designed to be deployed onto a web server. For further reading, consult: https://blogs.akamai.com/2013/10/web-shells-backdoor-trojans-and-rats.html

an organisation with malware that is configured to receive commands from a compromised web server. A system administrator may look at network traffic communicating with a trusted domain and dismiss it as legitimate.

Minimising risks and improving CMS security

7. The most common causes of CMS compromises are due to security oversights. Some of the most effective mitigations are listed below.

Mainstream Host

8. As an alternative to hosting and maintaining a CMS on your own infrastructure, consider using a managed CMS hosting service. Managed CMS hosting services maintain web infrastructure and content management applications, offering support and facilitating timely patching.

9. Government customers can use govCMS which is a hosting service for Drupal based websites. Further information can be found at www.govcms.gov.au.

10. For data that is not considered publically releasable, use an outsourced service that has been assessed, certified and accredited against the *Australian Government Information Security Manual* at the relevant classification level. Refer to ASD's Certified Cloud Services List at www.asd.gov.au.

Patch Management

11. A common cause of cyber intrusion is running an out-dated web server and CMS software. This makes exploitation of the CMS trivial in some instances. This risk can be minimised by:

    a. having an established process to test and deploy patches for the CMS software.

    b. patching the host operating system and third party applications, including themes, frameworks and libraries used by the CMS.

12. A CMS runs on a package of software known as a web stack. Additionally, organisations may employ third-party applications or custom site-specific code. All of these components (as shown in Figure 1) need to be patched, as one vulnerable component could compromise the security of the other layers.

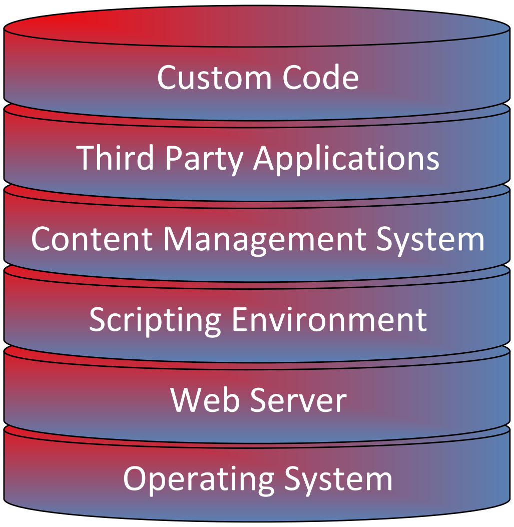

Figure 1: Components of a typical website

Vulnerability assessment of CMS installations

13. Use web vulnerability assessment tools to scan CMS installations for common vulnerabilities. Scanning tools are available on the internet, for example Arachni or CMS-specific tools such as WPScan for WordPress and the Security Review module for Drupal.

14. Conduct vulnerability assessments on custom code or modules that are used for CMS deployment.

Account Management

15. Poor management of legitimate access can lead to the compromise of a CMS. This risk can be minimised by:

    a. changing default usernames and passwords, including all related services, such as database access.

    b. using strong passwords / passphrases.

    c. ensuring passwords are stored by the CMS as salted hashes rather than cleartext. This prevents an adversary from obtaining cleartext passwords in the event of a compromise.

    d. restricting access to the administrator interface for the CMS from approved or internal IP addresses.

Hardening CMS installations

16. Use trusted, supported third-party plugins for the CMS.

17. Disable unnecessary functionality and plugins. By removing unneeded functionality you reduce the attack surface available to adversaries.

18. Disable or remove detailed debug or error messages in CMS webpages. Web pages that may disclose sensitive debug information, for example phpinfo() pages, should also be removed. An adversary could use this information to profile a CMS and identify out-dated software or plugins that may be vulnerable.

19. Version information that may be displayed by default on CMS webpages, for example in the page footer or in the META tags on each webpage, should be removed. Note, it is still possible to fingerprint the type and version of a CMS using automated tools such as BlindElephant[2].

20. Follow vendor advice on best practices for securing CMS installations. Refer to references below.

Monitoring CMS installations

21. Controls that aid in the detection of unauthorised modification of the website hosted on the CMS:

    a. using change management to manage the deployment of new versions of your website.

    b. using source control to manage development of custom code.

    c. using file integrity monitoring to manage and detect unauthorised changes to webpages. A properly configured file integrity monitor will assist in detecting modification of legitimate content and additions of malicious scripts.

22. Monitoring services that track compromised websites such as zone-h.org and xssed.com can be used to check if your site has been publicly defaced. These sites are limited in that they rely on

[2] https://community.qualys.com/community/blindelephant

users to report compromised websites, and hence generally only list public site defacements. It is highly unlikely that in the event that a CMS is compromised and used as command and control infrastructure it will be listed on these types of sites.

References

- Drupal: Securing your site: https://www.drupal.org/security/site-configuration
- WordPress – Hardening WordPress: http://codex.wordpress.org/Hardening_WordPress
- Joomla! Security Checklist: http://docs.joomla.org/Security_Checklist
- Open Web Application Security Project: https://www.owasp.org

Contact Details

23. Australian government customers with questions regarding this advice should contact ASD Advice and Assistance by emailing asd.assist@defence.gov.au or by calling 1300 CYBER1 (1300 292 371).

24. Australian businesses or other private sector organisations seeking further information should contact CERT Australia by emailing info@cert.gov.au or by calling 1300 172 499.