November 2015

Web Shells – Threat Awareness and Guidance

Overview

1. The purpose of this advisory is to highlight the frequent use of web shells as an exploitation vector. Web shells can be used to leverage unauthorised access and can lead to wider network compromise. This advisory outlines the threat and provides prevention, detection and mitigation strategies for administrators of web servers that have active content languages installed.
2. The ACSC has responded to multiple instances where the use of web shells by Advanced Persistent Threat (APT) and criminal groups has led to significant cyber incidents in Australia and globally.
3. This product was developed in collaboration with ACSC partners in the United Kingdom, United States, Canada and New Zealand based on activity seen targeting organisations across these countries. The detection and mitigation measures outlined in this document represent the shared judgement of all participating agencies.

Details

Web Shell Description

4. A web shell is a script that can be uploaded to a web server to enable remote administration of the machine. Infected web servers can either be internet-facing or internal to the network, where the web shell is used to pivot further to internal hosts.
5. A web shell can be written in any language that the target web server supports. The most commonly observed web shells are written in languages that are widely supported, such as PHP and ASP. Perl, Ruby, Python and Unix shell scripts are also used.
6. Using network reconnaissance tools, an adversary can identify vulnerabilities that can be exploited and result in the installation of a web shell. For example, these vulnerabilities can exist in content management systems (CMS) or web server software.
7. Once successfully uploaded, an adversary can use the web shell to leverage other exploitation techniques to escalate privileges and to issue commands remotely. These commands are directly linked to the privilege and functionality available to the web server and may include the ability to add, delete and execute files as well as the ability to run shell commands, further executables, or scripts.

How/Why are they used by malicious adversaries?

8. Web shells are frequently used in compromises due to the combination of remote access and functionality. Even simple web shells can have a considerable impact and often maintain minimal presence.

9. Web shells are utilised for the following purposes:

    i. Harvesting and exfiltration of sensitive data and credentials;

    ii. To upload additional malware for the potential of creating, for example, a watering hole for infection and scanning of further victims;

    iii. To use as a relay point to issue commands to hosts inside the network without direct internet access;

    iv. To use as command and control infrastructure, potentially in the form of a bot in a botnet or in support of compromises to additional external networks. This could occur if the adversary intends to maintain long-term persistence.

10. While a web shell itself would not normally be used for denial of service (DoS) attacks, it can act as a platform for uploading further tools, including DoS capability.

Examples

11. Web shells such as China Chopper, WSO, C99 and B374K are frequently chosen by adversaries; however these are just a small number of known used web shells. (Further information linking to IOCs and SNORT rules can be found in the Additional Resources section).

- ***China Chopper*** – A small web shell packed with features. Has several command and control features including a password brute force capability.
- ***WSO*** – Stands for “web shell by orb” and has the ability to masquerade as an error page containing a hidden login form.
- ***C99*** – A version of the WSO shell with additional functionality. Can display the server’s security measures and contains a self-delete function.
- ***B374K*** – PHP based web shell with common functionality such as viewing processes and executing commands.

Delivery Tactics

12. Web shells can be delivered through a number of web application exploits or configuration weaknesses including:

- Cross-Site Scripting;
- SQL Injection;
- Vulnerabilities in applications/services (e.g. Wordpress or other CMS applications);
- File processing vulnerabilities (e.g. upload filtering or assigned permissions);
- Remote File Include (RFI) and Local File Include (LFI) vulnerabilities;
- Exposed Admin Interfaces (possible areas to find vulnerabilities mentioned above).

13. The above tactics can be and are combined regularly. For example, an exposed admin interface also requires a file upload option, or another exploit method mentioned above, to deliver successfully.

Prevention and Mitigation

14. Installation of a web shell is commonly accomplished through web application vulnerabilities or configuration weaknesses. Therefore, identification and closure of these vulnerabilities is crucial to avoiding potential compromise. The following suggestions specify good security and web shell specific practices:

- Employ regular updates to applications and the host operating system to ensure protection against known vulnerabilities.
- Implement a least-privileges policy on the web server to:
  - Reduce adversaries' ability to escalate privileges or pivot laterally to other hosts.
  - Control creation and execution of files in particular directories.
- If not already present, consider deploying a demilitarised zone (DMZ) between your webfacing systems and the corporate network. Limiting the interaction and logging traffic between the two provides a method to identify possible malicious activity.
- Ensure a secure configuration of web servers. All unnecessary services and ports should be disabled or blocked. All necessary services and ports should be restricted where feasible. This can include whitelisting or blocking external access to administration panels and not using default login credentials.
- Utilise a reverse proxy or alternative service, such as mod_security, to restrict accessible URL paths to known legitimate ones.
- Establish, and backup offline, a "known good" version of the relevant server and a regular change-management policy to enable monitoring for changes to servable content with a file integrity system.
- Employ user input validation to restrict local and remote file inclusion vulnerabilities.
- Conducting regular system and application vulnerability scans is an effective method of establishing areas of risk. While this does not protect against zero day attacks it will highlight possible areas of concern.
- Furthermore, the deployment of a web application firewall, as well as regular virus signature checks, application fuzzing, code reviews and server network analysis can assist with establishing areas of risk and also help to minimise zero day exploits.

Detection

15. Due to the potential simplicity and ease of modification of web shells, they can be difficult to detect. For example, anti-virus products have been known to produce poor results in detecting web shells.

16. The following may be indicators that your system has been infected by a web shell. Note a number of these indicators are common to legitimate files. Any suspected malicious files should be considered in the context of other indicators and triaged to determine whether further inspection or validation is required.

- Abnormal periods of high site usage (due to potential uploading and downloading activity);
- Files with an unusual timestamp (e.g. more recent than the last update of the web applications installed);
- Suspicious files in internet accessible locations (web root);
- Files containing references to suspicious keywords such as cmd.exe or eval;

- Unexpected connections in logs. For example:
  - A file type generating unexpected or anomalous network traffic (e.g. a JPG file making requests with POST parameters);
  - Suspicious logins originating from internal subnets to DMZ servers and vice versa.
- Any evidence of suspicious shell commands, such as directory traversal, by the web server process.

17. Some web shells will display differently depending on the user-agent string. For example, the shell may not display to a search engine spider's user-agent. This can be an effective method of identification. To do this, there are plugins (such as "User-Agent Switcher") that can assist with temporarily changing a user-agent.

18. Client characteristics can also allude to possible web shell activity. One particular example that could be present in web access logs, is that the client will often visit only the web shell script URI itself whereas a standard user would be seen to load the webpage from a linked page/referrer or would be expected to load additional content/resources. In addition, performing frequency analysis on the web access logs could indicate the location of a web shell. Most legitimate URI visits will contain varying user-agents, whereas a web shell is generally only visited by the creator resulting in limited user-agent variants.

Summary

19. As a whole, web shells are commonly used by malicious adversaries, APT or criminal groups, due to the ease of which they can be deployed and their wide array of functionality. Due to the difficulty in detection, an adequate and resilient security posture is necessary and important to combat this potential threat.

Contact details

20. Australian government customers with questions regarding this advice should contact the ACSC on 1300 CYBER1 (1300 292 371) or asd.assist@defence.gov.au

21. 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.

Additional Resources

- *Securing Content Management Systems (CMS)*

  http://asd.gov.au/publications/protect/securing-cms.htm

- FireEye China Chopper – *The Little Malware That Could. Detecting and Defeating the China Chopper Web Shell*

  https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rptchina-chopper.pdf

- MANDIANT - *Old Web Shells New Tricks*

  https://www.owasp.org/images/c/c3/ASDC12-Old_Webshells_New_Tricks_How_Persistent_Threats_haverevived_an_old_idea_and_how_you_can_detect_them.pdf

- FireEye – *Breaking Down the China Chopper Web Shell Part I*

  https://www.fireeye.com/blog/threat-research/2013/08/breaking-down-the-chinachopper-web-shell-part-i.html

- FireEye – *Breaking Down the China Chopper Web Shell Part II*

  https://www.fireeye.com/blog/threat-research/2013/08/breaking-down-the-chinachopper-web-shell-part-ii.html

- WSO Information

  http://www.stratigery.com/phparasites/wso.html

- Exploit-db – *China Chopper*

  https://www.exploit-db.com/docs/27654.pdf

- C99

  www.cybersecurity.me/ids-intrusion-detection-systems-snort-bro/snort-ids-intrusiondetection-system/snort-signatures-rules-to-detect-webshells-c99-c99shell-and-more/

- INFOSEC Institute – Web Shell Detection

  Resources.infosecinstitute.com/web-shell-detection/

Traffic Light Protocol

22. The following table lists the classification levels used in the traffic light protocol (TLP) and describes the restrictions on access and use for each classification level.

TLP classification	Restrictions on access and use
RED	**Highly restricted** **Access to and use by your CERT Australia security contact officer(s) only** You must ensure that your CERT Australia security contact officer does not disseminate or discuss the information with any other person, and you shall ensure that you have appropriate systems in place to ensure that the information cannot be accessed or used by any person other than your CERT Australia security contact officer.
AMBER	**Restricted internal access and use only** Subject to the below, you shall only make AMBER publications available to your employees on a 'need to know basis' strictly for your internal purposes only to assist in the protection of your ICT systems. In some instances you may be provided with AMBER publications which are marked to allow you to also disclose them to your contractors or agents on a 'need to know basis' – strictly for your internal purposes only to assist in the protection of your ICT systems.
GREEN	**Restricted to closed groups and subject to confidentiality** You may share GREEN publications with external organisations, information exchanges, or individuals in the network security, information assurance or critical network infrastructure community that agree to maintain the confidentiality of the information in the publication. You may not publish or post on the web or otherwise release it in circumstances where confidentiality may not be maintained.
WHITE	**Not restricted** WHITE publications are not confidential. They contain information that is for public, unrestricted dissemination, publication, web-posting or broadcast. You may publish the information, subject to copyright and any restrictions or rights noted in the information.
NOT CLASSIFIED	Any information received from CERT Australia that is not classified in accordance with the TLP must be treated as AMBER classified information, unless otherwise agreed in writing by the Attorney-General's Department.