Web Applications: Supplement

# Notes and References

The tool phpMyAdmin has been available since 1998, and the major release dates are provided in Table 21-1.

Table 21-1. Release dates of major versions of phpMyAdmin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.3.9 | January 2011 |  | 4.3.0 | December 2014 |
| 3.4.0 | May 2011 |  | 4.4.0 | April 2015 |
| 3.5.0 | April 2012 |  | 4.5.0 | September 2015 |
| 4.0.0 | May 2013 |  | 4.6.0 | March 2016 |
| 4.1.0 | December 2013 |  | 4.7.0 | March 2017 |
| 4.2.0 | May 2014 |  |  |  |

(Source: <https://www.phpmyadmin.net/files/>)

Table 21-2 provides the default included version of phpMyAdmin for various distributions. CentOS includes phpMyAdmin in EPEL, so all the CentOS 7 systems have the same version of phpMyAdmin (which is updated); the same holds true for CentOS 6. Because CentOS 5 is no longer supported, EPEL is no longer updated for CentOS 5.

Table 21-2. Default included version of phpMyAdmin, by Linux distribution

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mint |  | 14 | 3.4.11 | 13.1 | 4.0.7 | 15.10 | 4.2.12 |
| 18.2 | 4.5.4 | 13 | 3.4.10 | 12.3 | 3.5.6 | 15.04 | 4.2.12 |
| 18.1 | 4.5.4 | 12 | 3.4.5 | 12.2 | 3.5.2 | 14.10 | 4.2.6 |
| 18 | 4.5.4 | 11 | 3.3.10 | 12.1 | 3.4.7 | 14.04 | 4.0.10 |
| 17.3 | 4.0.10 | 10 | 3.3.7 | 11.4 | N/A | 13.10 | 4.0.6 |
| 17.2 | 4.0.10 | OpenSuSE |  | Ubuntu |  | 13.04 | 3.5.8 |
| 17.1 | 4.0.10 | 42.3 | 4.7.1 | 17.10 | 4.6.6 | 11.04 | 3.3.10 |
| 17 | 4.0.10 | 42.2 | 4.4.15 | 17.04 | 4.6.6 | 12.10 | 3.4.11 |
| 16 | 4.0.6 | 42.1 | 4.4.5 | 16.10 | 4.6.4 | 12.04 | 3.4.10 |
| 15 | 3.5.8 | 13.2 | 4.2.10 | 16.04 | 4.5.4 | 11.10 | 3.4.5 |

Table 21-3 provides the release dates for major Joomla! releases.

Table 21-3. Release dates of major versions of Joomla!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.8.0 | September 2017 | 3.1.0 | April 2013 |  |
| 3.7.0 | April 2017 | 3.0.0 | September 2012 |  |
| 3.6.0 | July 2016 | 2.5.0 | January 2012 | EOL December 2014 |
| 3.5.0 | March 2016 | 1.7.0 | July 2011 | EOL February 2012 |
| 3.4.0 | February 2015 | 1.6.0 | January 2011 | EOL August 2011 |
| 3.3.0 | April 2014 | 1.5.26 | March 2012 | Last entry in 1.5.x |
| 3.2.0 | November 2013 |  |  |  |

(Sources: <https://docs.joomla.org/Category:Version_History>, <https://downloads.joomla.org/cms>)

Table 21-4 provides the release dates for major versions of WordPress

Table 21-4. Release dates of major versions of Wordpress

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4.9 | November 2017 | 4.2 | April 2015 | 3.5 | December 2012 |
| 4.8 | June 2017 | 4.1 | December 2014 | 3.4 | June 2012 |
| 4.7 | December 2016 | 4.0 | September 2014 | 3.3 | December 2011 |
| 4.6 | August 2016 | 3.9 | April 2014 | 3.2 | July 2011 |
| 4.5 | April 2016 | 3.8 | December 2013 | 3.1 | April 2011 |
| 4.4 | December 2015 | 3.7 | October 2013 | 3.0 | June 2010 |
| 4.3 | August 2015 | 3.6 | August 2013 |  |  |

(Sources: <https://wordpress.org/about/roadmap>)

### Metasploit Modules

These Metasploit modules can be used to attack WordPress plugins.

* auxiliary/admin/http/wp\_custom\_contact\_forms
* auxiliary/admin/http/wp\_easycart\_privilege\_escalation
* auxiliary/admin/http/wp\_symposium\_sql\_injection
* auxiliary/admin/http/wp\_wplms\_privilege\_escalation
* exploit/unix/webapp/wp\_advanced\_custom\_fields\_exec
* exploit/unix/webapp/wp\_ajax\_load\_more\_file\_upload
* exploit/unix/webapp/wp\_asset\_manager\_upload\_exec
* exploit/unix/webapp/wp\_creativecontactform\_file\_upload
* exploit/unix/webapp/wp\_downloadmanager\_upload
* exploit/unix/webapp/wp\_easycart\_unrestricted\_file\_upload
* exploit/unix/webapp/wp\_foxypress\_upload
* exploit/unix/webapp/wp\_frontend\_editor\_file\_upload
* exploit/unix/webapp/wp\_google\_document\_embedder\_exec
* exploit/unix/webapp/wp\_holding\_pattern\_file\_upload
* exploit/unix/webapp/wp\_inboundio\_marketing\_file\_upload
* exploit/unix/webapp/wp\_infusionsoft\_upload
* exploit/unix/webapp/wp\_lastpost\_exec
* exploit/unix/webapp/wp\_mobile\_detector\_upload\_execute
* exploit/unix/webapp/wp\_nmediawebsite\_file\_upload
* exploit/unix/webapp/wp\_optimizepress\_upload
* exploit/unix/webapp/wp\_photo\_gallery\_unrestricted\_file\_upload
* exploit/unix/webapp/wp\_phpmailer\_host\_header
* exploit/unix/webapp/wp\_pixabay\_images\_upload
* exploit/unix/webapp/wp\_platform\_exec
* exploit/unix/webapp/wp\_property\_upload\_exec
* exploit/unix/webapp/wp\_reflexgallery\_file\_upload
* exploit/unix/webapp/wp\_revslider\_upload\_execute
* exploit/unix/webapp/wp\_slideshowgallery\_upload
* exploit/unix/webapp/wp\_symposium\_shell\_upload
* exploit/unix/webapp/wp\_total\_cache\_exec
* exploit/unix/webapp/wp\_worktheflow\_upload
* exploit/unix/webapp/wp\_wpshop\_ecommerce\_file\_upload
* exploit/unix/webapp/wp\_wptouch\_file\_upload
* exploit/unix/webapp/wp\_wysija\_newsletters\_upload

Exercises

1. Modify the configuration so that phpMyAdmin runs only over HTTPS.
2. Verify that phpMyAdmin on XAMPP 1.8.0 uses basic authentication rather than form-based authentication. Write a script to perform a brute force attack on its password.
3. XAMPP 1.8.0 uses phpMyAdmin 3.5.2 with PHP 5.4.4. Perform the phpMyAdmin Authenticated Remote Code Execution via preg\_replace() against the phpMyAdmin installation on a Windows system running XAMPP 1.8.0. Does it succeed?
4. Perform the preg\_replace() attack against a system protected by a Snort intrusion detection system with the default rule set. What alerts (if any) fire?
5. Configure phpMyAdmin to restrict access using allow and deny rules. Compare the result returned when a user is not permitted to access the server because of an access rule to the result returned when a user provides the wrong credentials. What conclusions can an attacker draw?
6. Try the module auxiliary/scanner/http/phpmyadmin\_login.
7. Try the Metasploit modules auxiliary/scanner/http/joomla\_pages and auxiliary/scanner/http/joomla\_plugins. How useful are they?
8. Modify the brute force password attack scripts to pass their attacks though a Burp Suite proxy.
9. Try the module auxiliary/scanner/http/joomla\_bruteforce\_login. How well does it work?
10. Many editors save backup copies of edited files, often changing the end of the file name; for example, after editing the WordPress configuration file wp-config.php, a file wp-config.php~ may be present (this is the default behavior on CentOS 5.4 for example). Is the presence of this file detectable by wpscan? Is the file served by the web server? What are the security consequences, if any?
11. Select one of the Metasploit modules that can attack a WordPress plugin. Install a vulnerable version of the plugin, and run the exploit.