Roteiro 4 - Tecnologias Hacker

Por Victor Vergara

Pentest MetaExploitable

Para testar a segurança de um web server wordpress 4.8.2 com Apache 2.2.8 e PHP 5.2.4 foi feito um Pentest utilizando as ferramentas nikito, wpscan e OWASP ZAP a fim de detectar falhas na segurança do servidor.

O Teste detectou 10 falhas de segurança de médio ou baixo risco que facilitam um ataque de algum invasor via diferentes métodos e 1 falha de erro de possível falha de interpretação e display do conteúdo do site. Sendo que essas falhas de segurança podem ser consertadas com pequenas alterações ou atualizações das aplicações utilizadas.

Vulnerabilidades Encontradas

Utilizando o `wpscan --url [ip-alvo]/wordpress` foi possível descobrir vários dados sobre o servidor:

Headers

- **Ameaça:** Com os Headers é possível descobrir os serviços e as versões desses, assim um invasor poderia fazer ataques especificamente para essas versões dos serviços
- **Tipo de Detecção:** Passiva

```
[+] Headers
| Interesting Entries:
|  - Server: Apache/2.2.8 (Ubuntu) DAV/2
|  - X-Powered-By: PHP/5.2.4-2ubuntu5.10
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

XML-RPC

- **Ameaça:** Protocolo de encode em xml que utiliza HTTP, que é um protocolo vulnerável deixando possível um man in the middle attack.
- **Tipo de Detecção:** Agressiva
- **Solução:** Utilizar um protocolo seguro de comunicação e armazenar e mandar as senhas criptografadas.

```
[+] XML-RPC seems to be enabled: http://192.168.68.113/wordpress/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
|  - http://codex.wordpress.org/XML-RPC_Pingback_API
|  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
|  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
|  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
|  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

Readme do WordPress

- **Ameaça:** acesso ao Readme do WordPress acessível, podendo contar dados e descrições do servidor e seus conteúdos
- **Tipo de Detecção:** Agressiva
- **Solução:** Remover o arquivo

```
[+] WordPress readme found: http://192.168.68.113/wordpress/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
```

WP-Cron

- **Ameaça:** acesso ao wp-cron.php, que é necessário por manter os conteúdos do site no ar, mas pode sofrer ataques DDOS facilmente, assim sendo não deixar os conteúdos do server disponíveis
- **Tipo de Detecção:** Agressiva
- **Solução:** Desativar o wp-cron.php ou não utilizar o HTTP para realizar a sua comunicação

```
[+] The external WP-Cron seems to be enabled:
http://192.168.68.113/wordpress/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
|  - https://www.iplocation.net/defend-wordpress-from-ddos
|  - https://github.com/wpscanteam/wpscan/issues/1299
```

Utilizando o `nikto -h http://[ip-alvo]/wordpress` foi possível descobrir mais essas vulnerabilidades:

PHP Query String

- **Ameaça:** Quando se utiliza o Apache mod_cgid o php abilita o uso de argumentos que possibilitam o Invasor a pegar informações sensíveis, rodar códigos com privilégios no servidor ou causar um Denial of Service.
- **Tipo de Detecção:** Agressiva
- **Solução:** Fazer o Update do PHP

```
+ OSVDB-: /wordpress/?-s: PHP allows retrieval of the source code via
the -s parameter, and may allow command execution. See
http://www.kb.cert.org/vuls/id/520827
```

Utilizando OWASP ZAP foi possível descobrir mais essas vulnerabilidades:

Sem CSP ou Headers HTTP não apropriado

- **Ameaça:** Não se utiliza headers apropriados, assim possibilitando ataques XXS e data injection
- **Tipo de Detecção:** Passiva
- **Solução:** Configurar o Content-Security-Policy header

Content Security Policy (CSP) Header Not Set

Source	raised by a passive scanner (Content Security Policy (CSP) Header Not Set)
CWE ID	693
WASC ID	15
Reference	▪ https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy ▪ https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html ▪ http://www.w3.org/TR/CSP/ ▪ http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html ▪ http://www.html5rocks.com/en/tutorials/security/content-security-policy/ ▪ http://caniuse.com/#feat=contentsecuritypolicy ▪ http://content-security-policy.com/

Anti-clickjacking Headers

- **Ameaça:** Não utiliza CSP com X-frame-Options e não protege contra ataques ClickJacking
- **Tipo de Detecção:** Passiva
- **Solução:** Configurar o Content-Security-Policy header com X-Frame-Options HTTP headers

Missing Anti-clickjacking Header

Source	raised by a passive scanner (Anti-clickjacking Header)
CWE ID	1021
WASC ID	15
Reference	▪ https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Anti-CSRF Tokens

- **Ameaça:** Possível para ataques CSRF, onde se podem fazer submissões repetitivas
- **Tipo de Detecção:** Passiva
- **Solução:** Configurar o uso de de tokens anti-CSRF

Absence of Anti-CSRF Tokens

Source	raised by a passive scanner (Absence of Anti-CSRF Tokens)
CWE ID	352
WASC ID	9
Reference	▪ http://projects.webappsec.org/Cross-Site-Request-Forgery ▪ http://cwe.mitre.org/data/definitions/352.html

JavaScript Cross-Domain Inclusão de Arquivos

- **Ameaça:** Server utiliza Javascripts de domínios terceiros, que podem ser mudados externamentes sem a aprovação do admin do servidor, podendo causar mudanças no sistema do servidor, pegando dados, alterando banco de dados etc
- **Tipo de Detecção:** Passiva
- **Solução:** Utilizar scripts internos ou de partes confiáveis que um usuário não possa alterar

Cross-Domain JavaScript Source File Inclusion

Source	raised by a passive scanner (Cross-Domain JavaScript Source File Inclusion)
CWE ID	829
WASC ID	15

IP privado disponível

- **Ameaça:** O usuário tem acesso ao IP privado do servidor ao receber um resposta, dando uma informação extra ao possível atacante
- **Tipo de Detecção:** Passiva
- **Solução:** Remover o IP do body da resposta HTTP

Private IP Disclosure

Source	raised by a passive scanner (Private IP Disclosure)
CWE ID	200
WASC ID	13
Reference	▪ https://tools.ietf.org/html/rfc1918

X-Powered-By Leaks

- **Ameaça:** Alguns Headers X-Powers-by retornam X-Powers sobre frameworks e componentes, podendo ajudar um atacante pegando informações sobre aplicações e suas versões utilizadas no servidor
- **Tipo de Detecção:** Passiva
- **Solução:** Suprimir X-Powers headers

Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)

Source	raised by a passive scanner (Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s))
CWE ID	200
WASC ID	13
Reference	▪ http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx ▪ http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html

Sem X-Content-Type Headers

- **Ameaça:** Não há X-Content-type no header fazendo com que algumas versões de browser façam o display das informações da response erradas
- **Tipo de Detecção:** Passiva
- **Solução:** Setar Content-Type dos headers corretamente e X-Content-Type-Options como nosniff

X-Content-Type-Options Header Missing

Source	raised by a passive scanner (X-Content-Type-Options Header Missing)
CWE ID	693
WASC ID	15
Reference	▪ http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx ▪ https://owasp.org/www-community/Security_Headers