

# Capture The Flag 101

SecTalks SYD0x0b

Sydney, Australia

27 October 2015

# #!/whoami



Pedram Hayati

- PhD (ComSci), Bsc (IT eng.)
- Partner at elttam
- Founder SmartHoneypot
- Launched SecTalks non-profit meetups

# What

Capture the flag

# Capture The Flag (CTF)

- An IT security puzzle
- Topics
  - Computer security
  - Computer science
  - Networking
  - IT operation
- Objective: Find a way to get the flag in a limited time

# Challenges

## Web/Network

- Weaknesses in web applications/servers

## Forensics

- Finding the needle in the haystack

## Cryptography

- Weakness in cryptography, primitives or implementations

## Reverse engineering

- Exploring a binary data (static and dynamic analysis)

## Exploitation

- Write a working code

## Miscellaneous

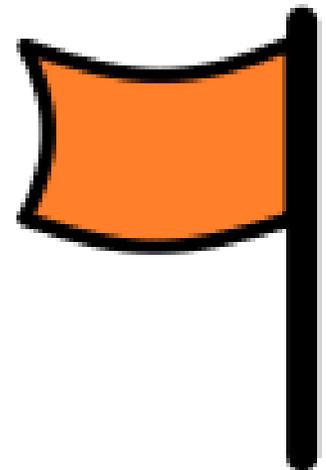
- Recon, Algorithm, Puzzle

# Flag

Hidden or seems impossible to access through normal ways

Types:

- A file
- A message
- A series of characters



# Solutions

- Writing a code snippet
- Exploiting a known vulnerabilities (or maybe zero-day)
- Script (e.g. bash)
- Using a **tool** (e.g. debugger, static code analysers, proxies)
- Finding an algorithm to get the flag
- Engineer a way to get to the flag

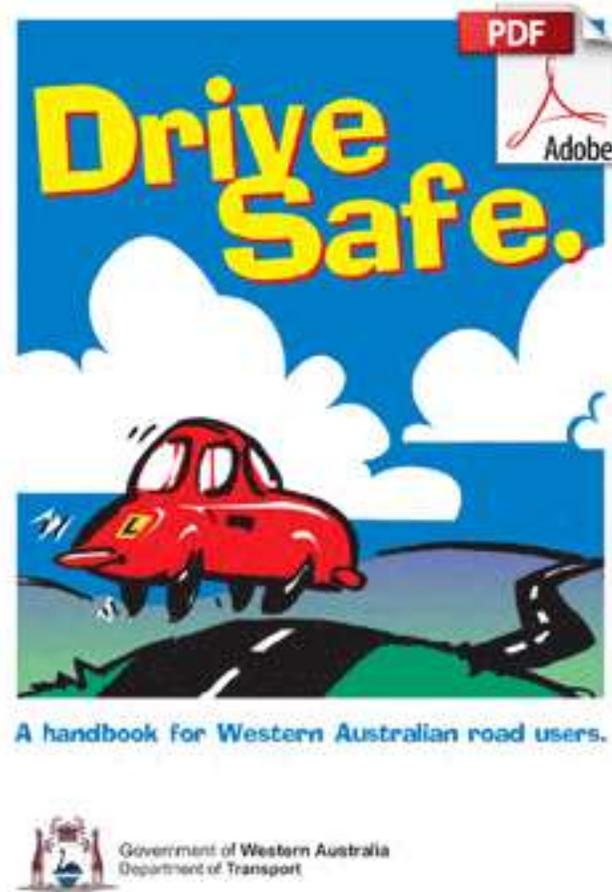


# Rules

- Bring along your gears.
- Attacking CTF scoring system results in disqualification.
- Attacking CTF competitors results in disqualification\*.
- You get point(s) for solving each challenge.
- Team with highest number of points wins.

# Why

Capture the flag



We never learn to drive by reading a book!

# We like games!

- We enjoy learning through games.
- We learn skillset as well as the theory.
- In a game we need to put the theory in practice.

# CTF

1. Understand the concept
2. Get hands-on skillset
3. Learn a new computer security techniques
4. Improve problem solving skills
5. Improves the thought process to think like a hacker/attacker
6. Helps to think out of the box and intuitively

# Do you want to be a security tester?

Security testing is not about running XYZ tool.

It is about thinking intuitively, out-of-the-box and coming up with edge cases that no one has thought.

You need to be able to think like a hacker/attacker.

# A taste of CTF

Get a taste of different CTF challenges

# Words Misheard

Silent! Listen...

I run to escape a persecution.

The eyes, they see.

The flag is insatiable.

Category: Recon, Points: 100

# Hint

Pay attention to the search result of “I run to escape a persecution”

# Hint

## Anagram

a word, phrase, or sentence formed from another by rearranging its letters: “Angel” is an anagram of “glean.”

# Solution

The flag is banalities that is one-word anagram for insatiable.

Reference

<http://hsctf.com/>

# Find the password

Browse to

<http://hax.tor.hu/warmup1/>

What is the password?

Category: Web, Points: 100

# Hint

View the source of the page source

Search for “Password:”

Search for “function a”

# Hint

- Press F12, go to Debugger, Click on line 398, Type a password, press Go.
- Press Ctrl+P, Type “\*thepw”.

# Solution

warmup1lolcopter

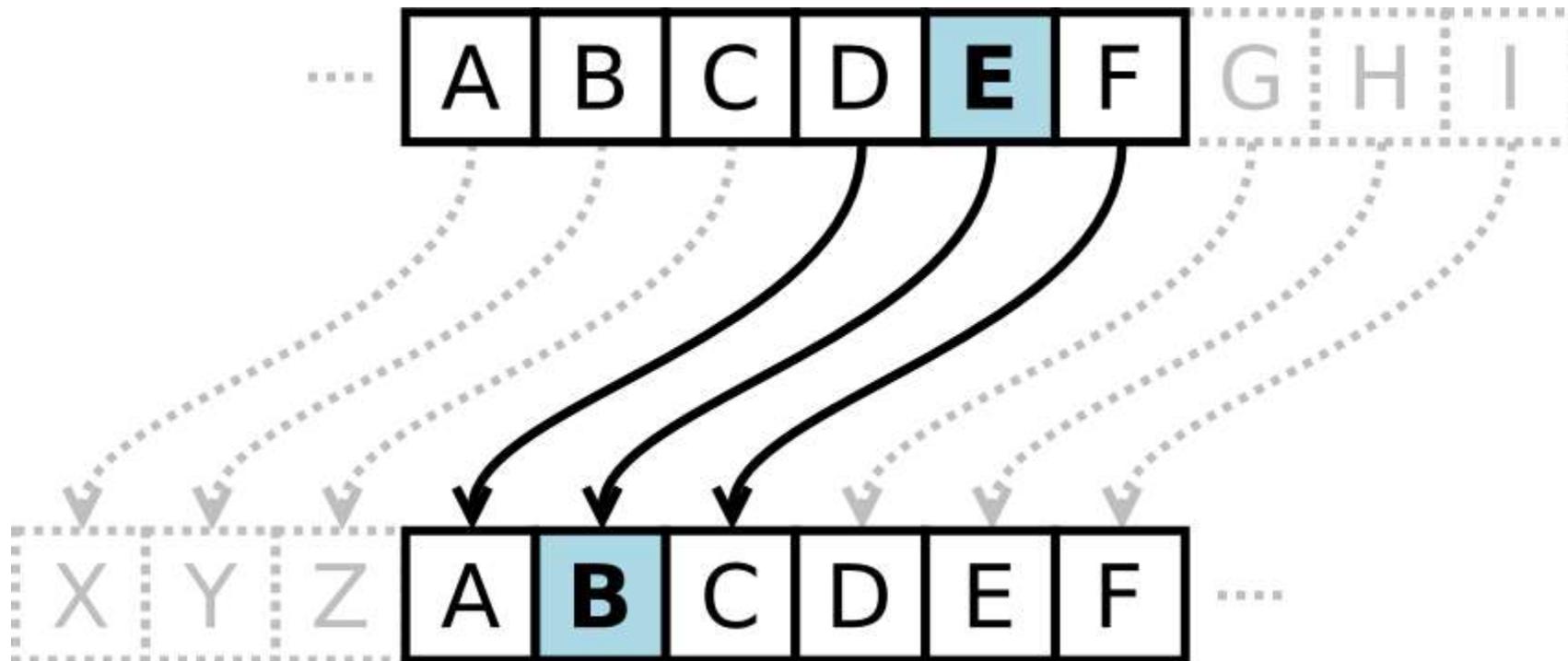
# What is the secret message?

GUR PNRFNE PVCURE VF BAR BS GUR FVZCYRFG RAPELCGVBA  
NYTBEVGUZF VA JUVPU RIREL YNGVA YRGGRE BS N TVIRA FGEVAT VF  
FVZCYL FUVSGRQ PLPYVNPYYL OL N PREGNVA BSSFRG. SBE  
PENPXVAT GUR RAPELCGVBA, JR PBHYQ VGRENGR BIRE NYY  
BCCBEGHAVGVRF NAQ NF BHE NYCUNORG HFRF WHFG 26 YNGVA  
YRGGREF, JR JBHYQ BOGNVA GUR QRPELCGRQ FGEVAT VA NG ZBFG 25  
GEVRF, JUVPU VF DHVGR GEVIVNY. GUR SYNT VF FRPGNYXF

Category: Crypto, Points: 100

# Hint

The Caesar cipher is one of the simplest encryption algorithms in which every latin letter of a given string is simply shifted cyclically by a certain offset.



# Hint

For cracking the encryption, we could iterate over all opportunities and as our alphabet uses just 26 Latin letters, we would obtain the decrypted string in at most 25 tries, which is quite trivial. An example of the Caesar cipher is rot13 (rotate by 13 places) in which the alphabet is rotated by exactly the halve alphabet.

# Solution

The Caesar cipher is one of the simplest encryption algorithms in which every Latin letter of a given string is simply shifted cyclically by a certain offset. For cracking the encryption, we could iterate over all opportunities and as our alphabet uses just 26 Latin letters, we would obtain the decrypted string in at most 25 tries, which is quite trivial.

The flag is sectalks

[http://www.cryptool-online.org/index.php?option=com\\_content&view=article&id=48&Itemid=95&lang=en](http://www.cryptool-online.org/index.php?option=com_content&view=article&id=48&Itemid=95&lang=en)

# Never login through HTTP

We tapped a line while someone logging in to a server. Can you tell if the person managed to login successfully?

Here is the capture file:

[http://www.sectalks.org/ctf101/01-the-basics/ctf101\\_the-basics\\_forensic.zip](http://www.sectalks.org/ctf101/01-the-basics/ctf101_the-basics_forensic.zip)

Category: Forensic, Points: 150

# Hint

File extensions (.XYZ) do not tell us anything.

In Linux, run 'file [FILENAME]' to understand the file type.

# Hint

Open the file with wireshark

# Hint

Find http communications

Right click and select “TCP Follow”

# Hint

The person tried to login twice.

# Solution

Filter: tcp.stream eq 1 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
114	10.145598000	::1	::1	TCP	88	55919→80 [SYN] Seq=0 W
116	10.145657000	::1	::1	TCP	88	80→55919 [SYN, ACK] S
118	10.145671000	::1	::1	TCP	76	55919→80 [ACK] Seq=1
120	10.145685000	::1	::1	TCP	76	[TCP Window Update] 8
121	10.146660000	::1	::1	HTTP	469	GET /totalsecureauth/
122	10.146684000	::1	::1	TCP	76	80→55919 [ACK] Seq=1
123	10.147517000	::1	::1	HTTP	479	HTTP/1.1 200 OK (tex
124	10.147541000	::1	::1	TCP	76	55919→80 [ACK] Seq=39
149	12.567047000	::1	::1	HTTP	555	GET /totalsecureauth/
151	12.567131000	::1	::1	TCP	76	80→55919 [ACK] Seq=40
153	12.567648000	::1	::1	HTTP	504	HTTP/1.1 200 OK (tex

# Solution

```
Stream Content
Accept-Encoding: gzip,deflate,sdch
Accept-Language: de,en;q=0.8
Cookie: guest=cb9e7954eb2bbc8ffa77f138a4d3e61a_1401788412

HTTP/1.1 200 OK
Date: Tue, 03 Jun 2014 09:40:14 GMT
Server: Apache/2.4.9 (Unix) PHP/5.5.11 OpenSSL/1.0.1g mod_perl/2.0.8-dev Perl/v5.16.3
X-Powered-By: PHP/5.5.11
Content-Length: 158
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html

<form method="post" target="login.php">user<input name="user">
password<input type="password" name="password">
  <input type="submit" value="Login">
</form>POST /totalsecureauth/login.php HTTP/1.1
Host: localhost
Connection: keep-alive
Content-Length: 23
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/35.0.1916.114 Safari/537.36
Content-Type: application/x-www-form-urlencoded
DNT: 1
Referer: http://localhost/totalsecureauth/login.php
Accept-Encoding: gzip,deflate,sdch
Accept-Language: de,en;q=0.8
Cookie: guest=cb9e7954eb2bbc8ffa77f138a4d3e61a_1401788412

user=test&password=testHTTP/1.1 200 OK
Date: Tue, 03 Jun 2014 09:40:19 GMT
Server: Apache/2.4.9 (Unix) PHP/5.5.11 OpenSSL/1.0.1g mod_perl/2.0.8-dev Perl/v5.16.3
X-Powered-By: PHP/5.5.11
Content-Length: 172
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
Content-Type: text/html

wrong password<form method="post" target="login.php">user<input name="user">
password<input type="password" name="password">
  <input type="submit" value="Login">
</form>
```

# Solution

Filter: tcp.stream eq 5

No.	Time	Source	Destination	Protocol	Length	Info
345	26.25675600	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
346	26.25681900	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
347	26.25683400	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
348	26.25685100	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
349	26.25700000	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
350	26.25703100	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
351	26.25770300	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
352	26.25773500	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
422	29.88030800	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
424	29.88035300	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
425	29.88103800	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
426	29.88106000	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
427	29.96609600	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
428	29.96612900	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
429	29.96687600	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
430	29.96690200	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
467	34.96832700	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
468	34.96838500	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
469	34.96839500	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
473	35.48257200	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000
475	35.48261500	:::1	:::1	TCP	60	55031 → 55031 [RST] Seq=1000000000

Stream Content

```
POST /totalsecureauth/login.php HTTP/1.1
Host: localhost
Connection: keep-alive
Content-Length: 28
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Origin: http://localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_0_0; rv:41.0) Gecko/20100101 Firefox/41.0
Content-Type: application/x-www-form-urlencoded
DNT: 1
Referer: http://localhost/totalsecureauth/login.php
Accept-Encoding: gzip, deflate, sdch
Accept-Language: de,en;q=0.8
Cookie: guest=cb9e7954eb2bbc8ffa77f138a4d3e61a_1401788888

user=admin&password=rapunzelHTTP/1.1 200 OK
Date: Tue, 03 Jun 2014 09:40:28 GMT
Server: Apache/2.4.9 (Unix) PHP/5.5.11 OpenSSL/1.0.1g
X-Powered-By: PHP/5.5.11
Set-Cookie: admin=a47cccd7dacec4c59dc2a4c4a943d9df_421
Set-Cookie: guest=deleted; expires=Thu, 01-Jan-1970 00:00:00 GMT
Content-Length: 175
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

you are now admin<form method="post" target="login.php"
password<input type="password" name="password">
<input type="submit" value="Login">
```

▶ Frame 350: 76 bytes on wire (60 bytes captured) on interface 0  
▶ Null/Loopback  
▶ Internet Protocol Version 6, Src: :::1, Destination: :::1  
▶ Transmission Control Protocol, Src Port: 55031, Dst Port: 55031

# How to master a CTF?

The golden rule

# Practice



Don't have time?  
Come to SecTalks more

Practice and practice even more!

# SecTalks

What

Monthly technical (in)security talks  
and hands-on challenges, no bullshit!

How

CTF and/or Presentation

When

Monthly

Where

Currently in Perth, Sydney and  
Brisbane

Connect

IRC

irc.sectalks.org:6697 (SSL) channel:  
#sectalks

Twitter: sectalks

Next meetup

Find your next local meetup at

[www.sectalks.org](http://www.sectalks.org)

# Wrap up

Conclusion

# Wrap up

- What CTF is
- Two very benefits of CTF
  - Improves the thought process to think like a hacker/attacker
  - Helps to think out of the box and intuitively
- Went through Recon, Crypto, Web, Forensic challenges
  - Learn fundamental method to solve CTF challenges
- The golden CTF rule
  - Practice and practice even more

# Hand picked resources

- Cryptography
  - [9 minutes video on cryptography 101](#) (by Prof. D. Brumley)
  - [Introduction to Cryptography](#) (by picoctf.com)
  - [Ciphers and tools to test](#) (from cryptool-online.org)
  - [Cryptography step-by-step exercises](#) (from Matasano)

# Next

What will be covered in the next workshop

# Coming up

## Cryptography

- Common ciphers
- Cryptanalysis
- Breaking the crypto code using Python
- Some crypto challenges



That's all for now.

Get in touch if you have any questions.

Twitter: pi3ch

Email: [pedram@elttam.com.au](mailto:pedram@elttam.com.au)

# Copyright

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Visit: <https://creativecommons.org/licenses/by-nc-sa/4.0/>