Capture The Flag Contest – How it works

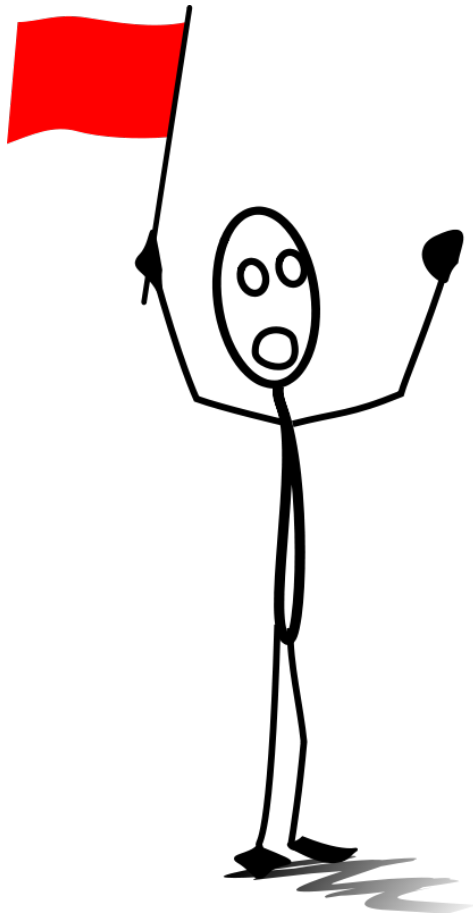
Benedikt Westermann

NTNU/Q2S

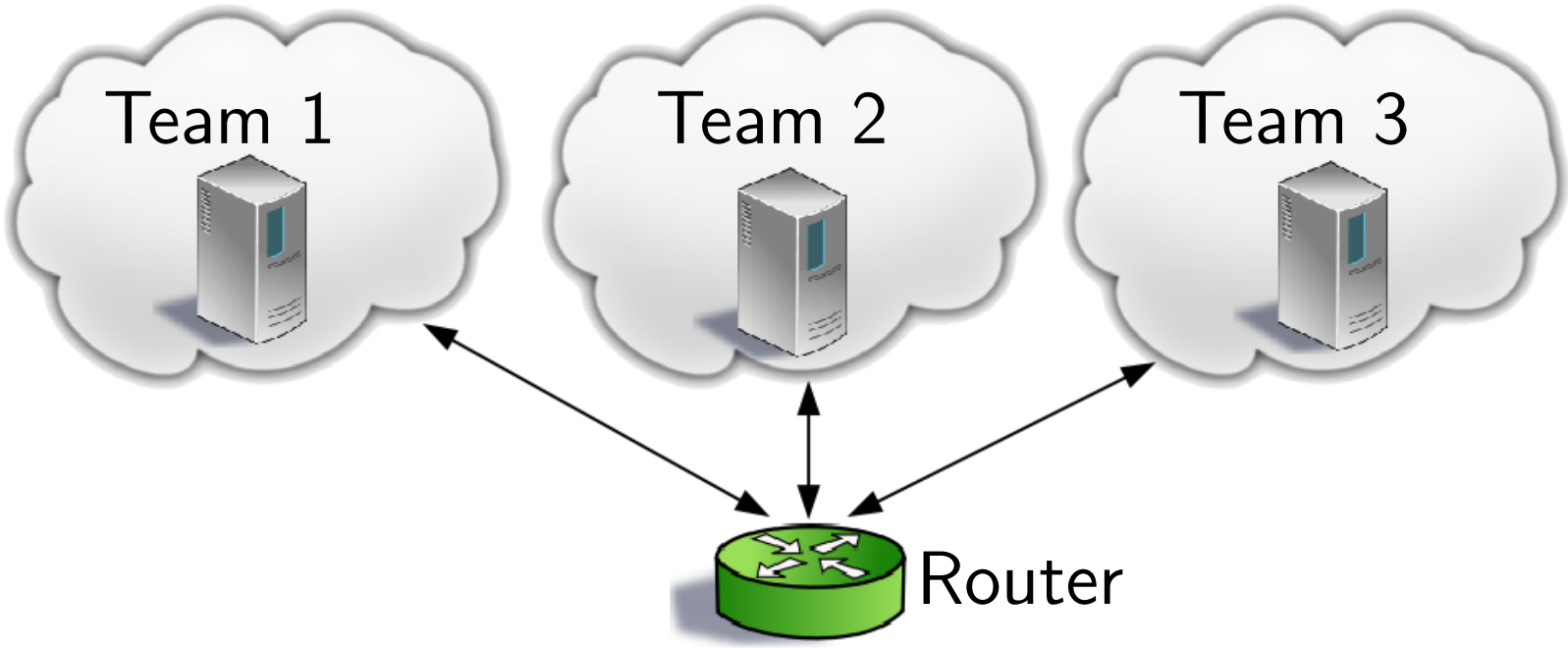
Norway, 16.06.2011

Idea of the Game

- several teams playing against each other
- each team has to protect its flag(s) from the other teams
- defending the flag gives you points
- capturing a flag of another team gives you points



The Basic Setting



Setup

- each team maintains a server
- the setup of the server is identical
- a server provides different services

Services on a Server

Services

- custom made
  - specifically written for a contest
  - no known vulnerabilities, e.g., CVE
- a service has various vulnerabilities

Example: Spammerd

```
SPAM Server
Version 4.5.3 (Testing)
201 SDP server ready
new john mypassword
200 OK
...
```

Objective

- find & remove vulnerabilities
- exploit them at the server of other teams to capture flags

What is a Flag?

- represents (valuable) data
  - needs to be protected
  - should not be leaked to others
  - is produced by a so-called gameserver
- a flag is 51 byte string
- a gameserver places flags on the team's server
  - every 20 seconds new flags are uploaded
  - old flags are retrieved

Example for a Flag

FLGgiQtQtEIXd1BdMI7JkAv9-i9vGTfaiDgLle7VZRx1aMcX9AR

How Do You Score?

- a flag captured from other teams gives you 1 point
- a flag retrieved by the gameserver gives you 1 point
  - the gameserver places a flag on your server
  - 20 seconds later it tries to retrieves the flag

  $\Rightarrow$ a broken service costs you *a lot* of points
  - how a flag is stored and retrieved depends on the service

Current team scores:

Team	Offensive	Defensive	Acquire	Aqs	Contribute
Unnamed1	0	22	Good	Good	Good
Unnamed2	0	22	Good	Good	Good
Siegen	0	22	Good	Good	Good

Example – Spammerd

1. The gameserver connects and stores a flag

```
Server: SPAM Server
Server: Version 4.5.3 (Testing)
Server: 201 SDP server ready
Client: new Dkwpsd aaabbb
Server: 200 OK
Client: to: xxxx
Server: 200 OK
Client: spam
Server: - Write your spam message here.
Server: - The first line is the subject
Server: - A line containing only a comma (.) will end the message
Client: FLGgiQtQtgIXd2GZcI7eeYoDuC8vGS_ZDK8rH6nAA2SwTfmsncH
Client: .
Server: 200 OK
Client: exit
Server: Bye Bye
```

Example – Spammerd

2. The gameserver connects and retrieves a flag

```
Server: SPAM Server
Server: Version 4.5.3 (Testing)
Server: 201 SDP server ready
Client: login DBFcyI cccddd
Server: 200 OK
Client: stat
Server: Mail currently in the sending queue:
Server: 0 - FLGgiQtQtgIXd2GZcI7eeYoDuC8vGS_ZDK8rH6nAA2SwTfmsncH
Server: 200 OK
Client: exit
Server: Bye Bye
```

Example – Spammerd

3. Your task: Steeling the flag

Challenge

- you don't know the correct username & password
- you know the source code (it's on your server)

Procedure

1. find (& fix) a vulnerability
2. find a way to exploit the vulnerability to extract the flag
3. write an exploit to extract the flag
4. submit the stolen flag to the gameserver

Some Examples for Vulnerabilities

Configuration

- passwords are not changed
- wrong access rights
- unnecessary services are running
- wrong/insecure configuration, e.g., apache

Vulnerabilities

- user input is not checked
- unintended functionality

Remote Command Execution

Example Code

```
$p = popen("find .. -name '*".$_GET['k']."*.h'",'r');
echo fread($p,1024)
```

- executes the find command (search for files)
- keyword `k` is determined via GET parameter
  (http://server/script.php?k=test)

Problem: Attacker sends k=t'; touch /tmp/hallo; echo '#

- find -name '*t'; touch /tmp/hallo; echo '#*.h'
- attacker can execute arbitrary commands

Path Traversal Attacks

Example Code

```
$file = fopen(‘‘/home/myservice/’’.$_GET[‘‘file’’]);
echo fread($file,60)
```

- shows the first 60 bytes of a file
- file is determined via GET parameter
  (http://server/script.php?file=myfile.txt)

Problem: Attacker sends file=../../../etc/passwd

- attacker gains unauthorized access, e.g., /etc/passwd
- attacker can open all files (the webserver can read)

Some Final Remarks

1. change all passwords as soon as possible
2. check for which services you can get points (and stop unnecessary services)
3. backup the original service files somewhere
4. check regularly the scoreboard
5. never trust *user data*! :-)

DON'T DO THIS

- block other teams
- perform a DOS attack on other machines
- delete important files, e.g., the services, or DB schemes

Service Overview

Acquire

- Web App in PHP

Aqs

- Web App in Python/Perl/C/ASM

Contribute

- Web App in Python

FTP

- FTP server written in C

Museum

- Web App in Perl

SpamAgent

- Web App written in C

SpamDB

- Web App written in PHP

SpamGen

- Web App written in python

SpamInOut

- Only binary available

Spammer

- Application written in Python

SHOF

- Tomcat App in Java

SyscallD

- Only a binary