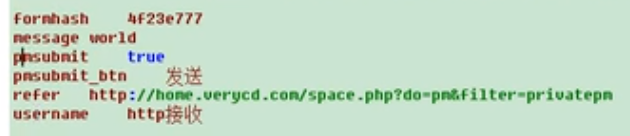
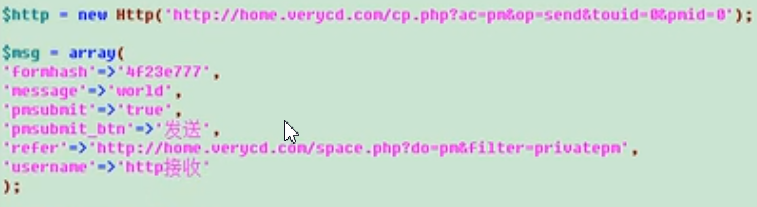
基于上节课的内容

1. 导包
2. 所要发送的地址



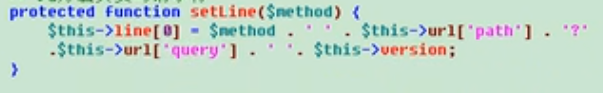
1. 调用post
2. 抓包
3. 抓取的包封装为数组



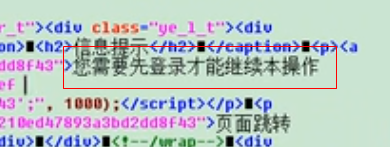
1. 调用post
2. 为了得到明显结果更改post



1. 调试缺url中的query，故更改setLine函数



1. 200ok后发现生成的html中



1. 加cookie



1. 发现还是不成功，分析可能是cookie与头信息有关联性，所以加上所有头信息





1. 成功

附：上节课整体代码

//http请求类的接口

interface Proto {

    //连接url

    function conn ($url);

    //发送get查询

    function get();

    //发送post查询

    function post();

    //关闭连接

    function close();

}

class Http implements Proto {

    const CRLF = "\r\n";

    protected $errno = -1;

    protected $errstr = '';

    protected $response = '';

    protected $url = null;

    protected $version = 'HTTP/1.1';

    protected $fh = null;

    protected $line = array();

    protected $header = array();

    protected $body = array();

    public function \_\_construct($url) {

        $this->conn($url);

        $this->setHeader('Host: ' . $this->url['host']);

    }

    //此方法负责写请求行

    protected function setLine() {

        $this->line[0] = $method . ' ' . $this->url["path"] . ' '. $this->version;

    }

    //此方法负责写头信息

    protected function setHeader($headerline) {

        $this->header[] = $headerline;

    }

    //此方法负责写主体信息

    protected funtion setBody() {

$this->body[] = http\_build\_query($body);

}

    //连接url

    public function conn ($url) {

        $this->url = parse\_url($url);

        //判断端口

        if(isset(this->url['port'])) {

            $this->url['port'] = 80;

        }

        $this->fh = fsockopen($this->url['host'], $this->url['port'], $this->errno, $this->errstr, 3);

    }

    //构造get请求的数据

    public function get() {

        $this->setLine('GET');

        $this->request();

        return $this->response;

    }

    //发送post查询

    public funtion post($body = array()) {

$this->setLine(‘POST’);

//设计content-type

$this->setHeader(‘Content-type:application/x-www-form-urlencoded’);

//构造主体信息，比GET不一样的地方

$this->setBody($body);

//计算content-length

$this->setHeader(‘Content-length:’ . strlen($this->body[0]));

$this->request();

}

    //真正请求

    public function request() {

        //把请求行，头信息，实体信息，翻盖一个数组里，便于拼接

$req = array\_merge($this->line, $this->header, array(''), $this->body,array(''))

        //print\_r($req);

        $req = implode(self::CRLF, $req);

        //echo $req; exit;

        fwrite($this->fh, $req);

        while(!feof($this->fh)) {

            $this->response .= fread($this->fh),1024

        }

        $this->close(); //关闭连接

        return $this->response;

    }

    //关闭连接

public function close() {

Fclose($this->fh);

    }

/\*

$url = 'https://www.bilibili.com/video/BV1js411g7Fw?p=3';

$http = new Http($url);

echo $http->get();

Set\_time\_limit(0);

$url = ‘...’;

for($i=1,$i<100;$i++) {

$str = str\_shuffle(‘...’);

$tit = substr($str,0,5);

$con = substr($str,6,8);

$http = new Http($url);

echo $http->post(array(‘tit’=>$,tit’con’=>$con,’submit’=>’留言’));

echo $tit,’------------’,$con,’<br />’;

usleep(2000);

\*/