|  |  |
| --- | --- |
| ICS | 35.030 |
| CCS | L 80 |

中华人民共和国国家标准

GB/T XXXXX—XXXX



信息安全技术 漏洞测试脚本及扫描插件适配接口规范

Information security technology-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(点击此处添加与国际标准一致性程度的标识)

（本草案完成时间：2023年10月）

XXXX - XX - XX发布 XXXX - XX - XX实施



目次

[前言 III](#_Toc15781)

[1 范围 4](#_Toc25830)

[2 规范性引用文件 4](#_Toc6183)

[3 术语和定义 4](#_Toc10310)

[3.1 4](#_Toc19313)

[3.2 4](#_Toc13256)

[3.3 4](#_Toc24559)

[4 缩略语 4](#_Toc23526)

[5 漏洞扫描任务执行流程 4](#_Toc5863)

[6 漏洞测试脚本和扫描插件接口规范 5](#_Toc2214)

[6.1 概述 5](#_Toc31958)

[6.2 插件信息 5](#_Toc28395)

[6.3 漏洞信息 6](#_Toc27627)

[6.4 测试输入 7](#_Toc32161)

[6.5 测试过程 7](#_Toc26379)

[6.5.1 模块初始化函数 7](#_Toc25880)

[6.5.2 测试准备函数 7](#_Toc14915)

[6.5.3 测试运行函数 7](#_Toc4666)

[6.5.4 测试清理函数 7](#_Toc29598)

[6.5.5 模块销毁函数 8](#_Toc1162)

[6.6 测试结果 8](#_Toc8106)

[7 漏洞测试脚本和扫描插件功能库 8](#_Toc23826)

[7.1 概述 8](#_Toc20279)

[7.2 功能类别 8](#_Toc13724)

[7.2.1 基础功能 8](#_Toc22559)

[7.2.2 协议支持 8](#_Toc2803)

[7.2.3 工具支持 8](#_Toc4360)

[7.3 示例 9](#_Toc6663)

[附录A （规范性） 10](#_Toc2179)

[参考文献 11](#_Toc20244)

1. 前言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件由全国信息安全标准化技术委员会（SAC/TC 260）提出并归口。

本文件起草单位：

本文件主要起草人：

信息安全技术 漏洞测试脚本及扫描插件适配接口规范

* 1. 范围

本规范规定了漏洞扫描器插件集成标准，以实现不同漏洞扫描器插件的统一管理、调用和结果融合，包括漏洞测试脚本的描述规范和扫描插件集成接口描述规范。

本规范适用于漏洞检测类产品。

* 1. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 25069-2022 信息安全技术 术语

* 1. 术语和定义

下列术语和定义适用于本文件。



**计算机信息系统 computer information system**

由计算机及其相关的和配套的设备、设施(含网络)构成的，按照一定的应用目标和规则对信息进行 采集、加工、存储、传输、检索等处理的人机系统。



**安全漏洞 vulnerability**

计算机信息系统在需求、设计、实现、配置、运行等过程中，有意或无意产生的缺陷。这些缺陷以不同形式存在于计算机信息系统的各个层次和环节之中， 一旦被恶意主体所利用，就会对计算机信息系统的安全造成损害，从而影响计算机信息系统的正常运行。



**漏洞扫描器 vulnerability scanner**

一种专注于主动检测系统安全漏洞的工具，通过遵循特定策略，模拟攻击者的行为，对目标系统进行全面扫描，及时发现并验证潜在的安全漏洞，并生成检测报告，提供相应的改进建议，以保障系统的安全性和稳定性。

* 1. 缩略语

下列缩略语适用于本文件。

CNNVD:中国国家信息安全漏洞库（China National Vulnerability Database of Information security）

CNVD:国家信息安全漏洞共享平台（China National Vulnerability Database）

CVE:公共漏洞和威胁暴露（Common Vulnerabilities& Exposures）

* 1. 漏洞扫描任务执行流程

漏洞扫描任务执行流程如下：

1)系统部署接入：用户部署漏洞扫描器，完成安装、配置等操作。

2)用户指定扫描目标：用户指定扫描的目标，包括目标的IP地址、端口号、以及访问URI路径、请求方法、账户名、密码等可选参数；目标IP地址可以是单一的IP地址、域名、或一个IP地址范围。

3)用户指定扫描策略：用户定义扫描策略，包括扫描速度、并发数等。

4)用户指定扫描插件：用户选择扫描过程需要使用的漏洞检测插件。

5)扫描任务执行：扫描器执行扫描任务，按照用户定义的策略，针对目标系统进行扫描。

4)结果分析和报告生成：扫描器收集扫描结果，包括发现的漏洞信息，汇总并生成报告。报告通常包括漏洞的等级、影响程度、建议修复措施等详细信息。

示例图见图 1。

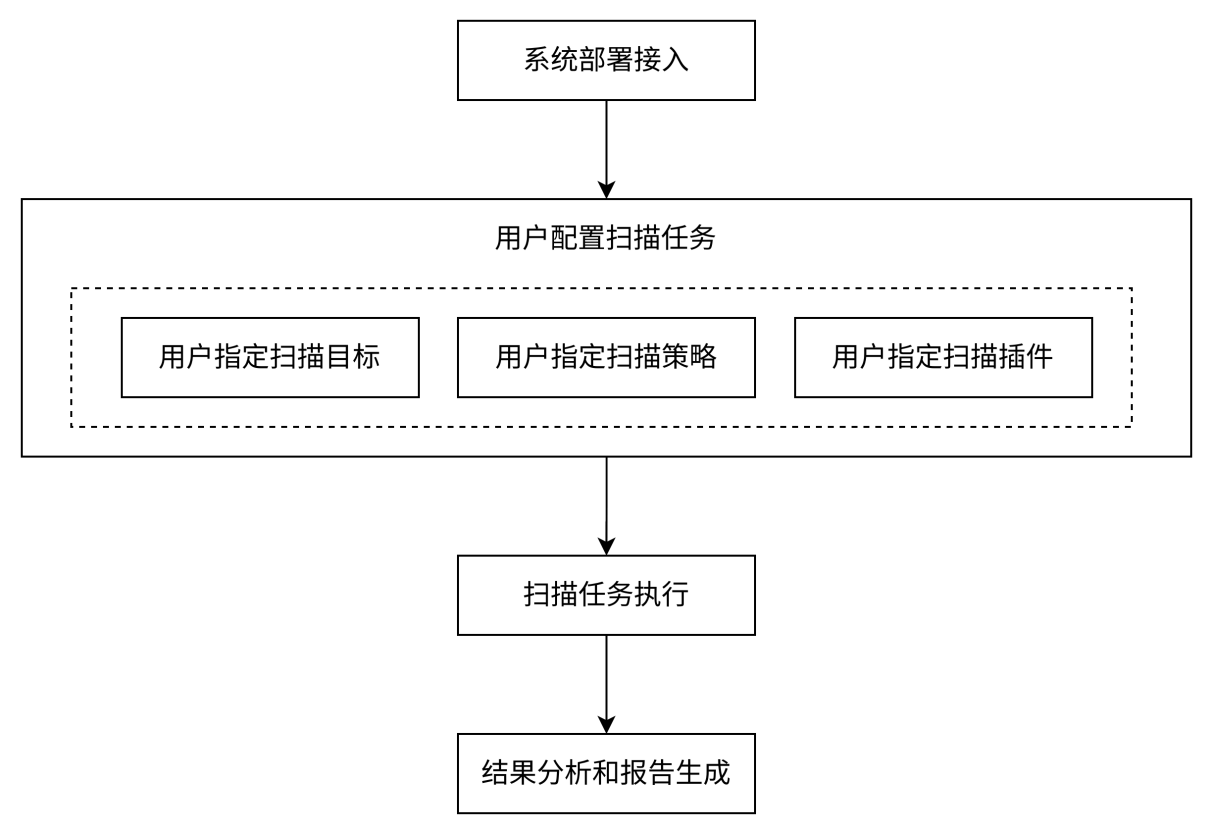


图 1 漏洞扫描任务执行示例图

* 1. 漏洞测试脚本和扫描插件接口规范
     1. 概述

漏洞扫描器的目标是挖掘目标系统上的漏洞风险，不同的漏洞的扫描检测方式各有差异，但建立统一的接口规范，一方面便于漏洞扫描器对不同扫描插件进行统一管理，也便于安全测试人员对新出现的漏洞编写合适的扫描插件，从而提升整体的安全响应能力。漏洞测试脚本和扫描插件接口规范包含下列五类信息：插件信息、漏洞信息、测试输入、测试过程、测试结果。

* + 1. 插件信息

“插件信息”是指该扫描插件自身相关的标识与描述信息，用于对漏洞扫描插件的整体分类与管理。定义这部分信息旨在提供插件的检索能力和去重能力。

扫描插件应该包括的信息字段有：名称、标识、描述、版本、作者、平台、架构、注释、类别、标签，具体含义见表 1。

表 1 插件信息字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 名称 | Name | 插件的名称，通常会简洁地描述插件的功能。 |
| 标识 | Id | 插件的唯一标识 |
| 描述 | Description | 对插件功能的详细描述 |
| 版本 | Version | 该插件的版本号信息，用于追踪和管理插件的更新和变更 |
| 作者 | Author | 创建此插件的人或团队的名称 |
| 平台 | Platform | 该插件针对的目标测试对象的平台信息，如Windows、Linux、Mac等 |
| 架构 | Arch | 该插件针对的目标测试对象架构信息，如x86\_64、aarch64等 |
| 注释 | Notes | 该插件有关的，提供给该插件的使用者的提示信息，通常是对插件输入参数的一些提示和注意事项 |
| 类别 | Classification | 该插件所在的分类类别，该字段被定义为分层结构，例如“linux/ssh”表示这个插件归属于“linux”分类下的“ssh”子分类 |
| 标签 | Tags | 该插件的标签信息，每个插件可以有多个标签，以列表形式组织 |

* + 1. 漏洞信息

“漏洞信息”是指该扫描插件关联的待测试漏洞信息，这些信息一方面可以供测试人员了解测试结果，还可以让测试人员进行有针对性的测试。漏洞信息应该包含的字段有：漏洞描述、CVSS评估、CVSS评分、CVE编号、CWE分类、CPE标识、发布信息、相关编号、引用信息、解决方案，具体含义见表 2。

表 2 漏洞信息字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 漏洞描述 | description | 漏洞印象范围及可能产生的威胁，至少包括漏洞类型、根本原因和造成影响中的一类信息，描述语言必须为中文，例如“Python 3.x到3.9.1在XXX. c中有一个缓冲区溢出，这可能导致在某些接受浮点数作为不可信输入的Python程序中远程执行代码” |
| CVSS评估 | cvss-metrics | 漏洞对应的CVSS评估的维度信息，例如“CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H” |
| CVSS评分 | cvss-score | 漏洞对应的CVSS评分信息，例如“9.8” |
| CVE编号 | cve-id | 漏洞对应的CVE编号信息，例如“CVE-2021-3177” |
| CWE分类 | cwe-id | 漏洞对应的漏洞分类信息，若存在多个可顺序给出，中间以逗号分隔，例如“CWE-120, CWE-118” |
| CPE标识 | cpe-id | 漏洞影响的产品的标识符，若存在多个可顺序给出，中间以逗号分隔，例如“cpe:2.3:a:python:python:\*:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:o:fedoraproject:fedora:32:\*:\*:\*:\*:\*:\*:\*” |
| 发布信息 | release | 漏洞发布的相关信息，包括发布者、发布平台和发布时间，例如“张三（组织A），于2019-01-01在XXX平台提交漏洞YYY” |
| 相关编号 | other-id | 漏洞在不同组织中的编号，例如，CNVD编号、CNNVD编号或其他组织自定义的漏洞编号等，若存在多个可顺序给出，中间以逗号分隔，例如“CVE-2021-3177，CNNVD-202101-1467”。 |
| 引用信息 | reference | 漏洞发布的信息来源或相关的参考资料，可以包括厂商公告、研究论文、博客文章等，以url形式记录，例如“https://\*\*\*.\*\*\*.\*\*\*”。 |
| 解决方案 | solution | 漏洞的现有解决方案或建议的缓解措施，例如“尽快升级Python版本到3.9.1以上” |

* + 1. 测试输入

“测试输入”是插件在运行前需要由用户指定的输入参数字段，通常定义了测试目标对象的标识信息如目标的IP地址、端口号等，以及插件运行的额外配置选项。插件应当在其加载生命周期的某个阶段向漏洞扫描器声明其所接受的额外配置选项的字段标识以及字段描述。相关的字段有：目标IP、目标端口、请求路径。注意这些字段是与插件的内容强相关的，因此只作为可选项，具体定义见表 3。

表 3 测试输入字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 目标IP | ip | 目标IP地址可以是单一的IP地址、IP地址范围、域名，例如“192.168.1.2” |
| 目标端口 | port | 该插件所测试的服务在目标上暴露的端口号，例如“8889” |
| 请求路径 | path | 该插件所测试服务请求的路径前缀，例如“/api” |

* + 1. 测试过程

“测试过程”是指该扫描插件针对目标系统展开测试的具体过程。在扫描插件中以函数代码的形式来定义扫描插件的测试过程，在这种形式下，插件开发人员拥有高度灵活的自定义能力，能够以代码形式实现任意复杂的测试过程。

为了能够与漏洞扫描器的其他部分协作，还需要定义漏洞扫描插件提供的统一接口函数，这些函数在插件的生命周期中的各个阶段被漏洞扫描器所调用。

* + - 1. 模块初始化函数

“模块初始化函数”在插件被漏洞扫描器加载到系统中后被调用，且只被调用一次。漏洞扫描器需要向该函数中传入一个上下文对象作为参数。漏洞扫描器利用该上下文对象暴露的函数功能，实现与漏洞扫描器的交互。例如，该上下文对象将提供一个注册函数，允许模块将上文中提到的插件信息和漏洞信息作为该模块的附加信息注册到漏洞扫描器中。

* + - 1. 测试准备函数

“测试准备函数”在“测试运行函数”被调用前执行。这通常意味着用户操作扫描器使用该插件开始一次扫描。与模块初始化函数类似，该函数接受一个上下文对象作为参数。在该函数中，插件开发者可以执行一些准备操作，例如，检查用户配置的测试输入是否满足最低要求，允许环境是否满足要求，初始化协议客户端等。

* + - 1. 测试运行函数

“测试运行函数”是模块运行的最核心代码，对应了用户实施的一次扫描动作。在该函数中，模块实现对应的扫描操作，如发送HTTP请求并获得响应、建立到测试对象的TCP连接并检查响应等。

与模块初始化函数类似，该函数接受一个上下文对象作为参数。通过该上下文对象，开发人员可以访问用户先前指定的测试输入信息，以及扫描插件提供的功能函数。此外，扫描结果也将通过调用上下文对象提供的一个统一的结果设置函数来返回给漏洞扫描器。

注意“测试运行函数”为同步函数而非异步函数，在该函数结束前，需要完成扫描过程。漏洞扫描器应该假设该函数在执行扫描的过程可能会持续较久时间，因此漏洞扫描器应当处理该情况并且防止用户界面的阻塞发生。

* + - 1. 测试清理函数

“测试清理函数”在“测试运行函数”被调用完成后执行。该函数对应着一次扫描任务的结束。与模块初始化函数类似，该函数接受一个上下文对象作为参数。在该函数中，插件开发者可以执行一些清理动作，例如，关闭多余的TCP连接，清理临时文件等。

* + - 1. 模块销毁函数

“模块销毁函数”在模块被整体从漏洞扫描器中卸载时被调用。与模块初始化函数类似，该函数接受一个上下文对象作为参数。在该函数中，开发者可以做一些模块级别的清理动作。

* + 1. 测试结果

“测试结果”是指模块在执行测试任务后得出的扫描结果，应当至少包含验证结果、目标信息、附加信息，如表 4所示。

表 4 测试结果字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 验证结果 | result | 明确指出该插件所扫描的漏洞是否存在，该字段应为True或者False |
| 目标信息 | target\_info | 该插件在测试过程中所搜集到的关于目标的信息，如IP地址、开放端口、探测到的服务版本号等，该字段的值应为一组键值对 |
| 附加信息 | additional | 在“验证结果”之外附加的信息，该字段记录到报告中用于提示测试人员 |

* 1. 漏洞测试脚本和扫描插件功能库
     1. 概述

漏洞测试脚本和扫描插件功能库是指由漏洞扫描器运行环境向扫描插件提供的功能库，目的是让扫描插件的编写者能够方便地实现扫描功能，简化编写的复杂性，同时减少对外部运行环境的依赖。漏洞扫描器需要向漏洞测试脚本提供的功能主要分类为：基础功能、协议支持、工具支持。

针对每一种不同的子功能模块，需要提供的函数接口内容也各有差异，漏洞扫描器应当将同一子功能模块下的内容以“类”或“包”或“命名空间”的形式进行组织，并提供相应的API接口文档，方便开发者进行查找。

* + 1. 功能类别
       1. 基础功能

“基础功能”是指漏洞扫描器向插件提供的一些基础的功能模块和功能函数，这部分库承载了漏洞扫描器可能使用到的通用功能，例如：日志模块、TAR压缩文件处理模块、JSON字符串解析模块、INI配置文件解析模块等。

* + - 1. 协议支持

“协议支持”是指漏洞扫描器向插件提供的一些常见协议的客户端和服务端实现功能模块和功能函数。由于在扫描过程中，插件可能需要用特定协议与测试目标进行交互，将协议的实现作为库可以简化插件开发。一些常见的协议例如：HTTP协议、DNS协议、TELNET协议、SSH协议等。

* + - 1. 工具支持

“工具支持”是指漏洞扫描器向插件提供安全领域的一些常用的工具的调用接口。由于在扫描过程中，插件可能需要借助外部工具完成特定任务，或进一步深入挖掘目标。因此有必要对外部工具进行包装。一些常见的工具有：HTTP协议、DNS协议、TELNET协议、SSH协议等。

* + 1. 示例

Metasploit扫描插件及Eternal blue漏洞测试脚本示例：

* + - 1. 漏洞扫描任务执行流程

系统部署接入

**1、系统环境：**Ubuntu20.04

**2、部署方法：**

参考[MSF官方文档](https://docs.metasploit.com/docs/using-metasploit/getting-started/nightly-installers.html)

使用如下命令安装MSF框架：

curl <https://raw.githubusercontent.com/rapid7/metasploit> omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall && \

chmod 755 msfinstall && \

./msfinstall

用户配置扫描任务

**1、指定扫描目标：**

扫描目标为处于同一网段下ip为192.168.75.132对应的主机。

**2、制定扫描策略：**

对该漏洞的扫描测试策略主要分为ARP存活探测扫描、SMB漏洞扫描以及Eternalblue载荷攻击三部分。

**3、指定扫描插件：**

本测试实例中指定扫描插件为Metasploit框架。

**4、执行扫描任务**

**5、分析结果并生成报告**

* + - 1. 插件信息字段说明

表 1 插件信息字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 名称 | Name | Metasploit框架（Metasploit Framework）：一个用于开发、测试和执行漏洞利用的开源工具，支持广泛的攻击场景，包括漏洞扫描、漏洞利用、后渗透测试等。 |
| 标识 | Id | \ |
| 描述 | Description | Metasploit框架（Metasploit Framework，简称MSF）是一个功能强大的开源渗透测试平台，广泛用于信息安全领域的漏洞开发、漏洞利用、渗透测试和网络防御。Metasploit框架由Rapid7公司维护和支持，最初由H. D. Moore于2003年创建。它为安全研究人员和渗透测试者提供了一个集成的环境，能够自动化测试、管理漏洞以及模拟攻击场景，帮助用户评估和加强网络安全防御。 |
| 版本 | Version | 6.3.30 |
| 作者 | Author | H. D. Moore |
| 平台 | Platform | Windows、Linux、Mac等。 |
| 架构 | Arch | x86、x86\_x64、ARM、MIPS、PowerPC、S390x等。 |
| 注释 | Notes | 1、msfupdate：确保Metasploit框架及其模块保持最新。更新可以提供最新的漏洞利用、修复和改进。  2、show options：检查模块的必需配置参数，并确保正确设置这些参数，未正确配置的参数可能导致攻击失败或未能成功建立会话。  3、show payloads：根据目标系统的架构和操作系统选择合适的payload、例如，选择windows/meterpreter/reverse\_tcp用于Windows系统，linux/x86/meterpreter/reverse\_tcp用于32位Linux系统。 |
| 类别 | Classification | \ |
| 标签 | Tags | \ |

* + - 1. 漏洞信息说明

表 2 漏洞信息字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 漏洞描述 | description | **漏洞类型**：永恒之蓝漏洞（CVE-2017-0144）是一种远程代码执行漏洞。它影响了Windows操作系统中的Server Message Block（SMB）协议，特别是SMBv1服务。  **根本原因**：该漏洞的根本原因是SMBv1协议在处理特制请求时存在一个缓冲区溢出漏洞。具体来说，当攻击者向易受攻击的Windows系统上的SMBv1服务发送特制的数据包时，系统无法正确处理这些数据包，从而导致缓冲区溢出。这种溢出可能允许攻击者在目标系统上执行任意代码。  **造成的影响**：永恒之蓝漏洞可能导致攻击者在远程计算机上执行任意代码。一旦成功利用该漏洞，攻击者可以获得目标计算机的控制权，从而可能导致系统的完全控制、数据泄露或系统崩溃。由于SMBv1协议的广泛使用和该漏洞的普遍存在，攻击者可以在不需要物理接触的情况下远程攻击大量计算机，从而使得这一漏洞成为一个严重的安全威胁。 |
| CVSS评估 | cvss-metrics | CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H |
| CVSS评分 | cvss-score | 8.1 |
| CVE编号 | cve-id | CVE-2017-0144 |
| CWE分类 | cwe-id | CWE-20、CWE-119、CWE-200、CWE-787 |
| CPE标识 | cpe-id | cpe:2.3:a:microsoft:server\_message\_block:1.0:*:*:*:*:*:*:\*  cpe:2.3:o:microsoft:windows\_10:-:*:*:*:*:*:*:\*  cpe:2.3:o:microsoft:windows\_7:-:sp1:*:*:*:*:*:* |
| 发布信息 | release | Microsoft（微软公司），于2017-03-14在Microsoft TechNet平台发布了漏洞MS17-010。此漏洞被公开为微软的安全公告，主要针对Windows操作系统中的SMB协议存在的多个远程代码执行漏洞。 |
| 相关编号 | other-id | CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147, CVE-2017-0148, CNNVD-201703-331, CNVD-2017-02798。 |
| 引用信息 | reference | **1、微软安全公告（MS17-010）**: <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2017-0143>  **2、EternalBlue漏洞分析（博客文章）**: https://www.rapid7.com/blog/post/2017/05/14/eternalblue-exploit-analysis/  **3、NSA工具泄露事件分析（博客文章）**: https://blog.trendmicro.com/trendlabs-security-intelligence/analyzing-the-evasion-techniques-used-in-the-eternalblue-exploit/。 |
| 解决方案 | solution | **1、安装微软官方补丁**: 尽快应用微软发布的MS17-010安全补丁。该补丁修复了受影响的Windows操作系统中的SMBv1漏洞。补丁可以通过Windows Update或从微软官方网站下载并安装。  [MS17-010补丁下载链接](https://portal.msrc.microsoft.com/en-us/security-guidance/advisory/CVE-2017-0143)：https://msrc.microsoft.com/en-us/security-guidance/advisory/CVE-2017-0143  **2、禁用SMBv1协议**: 如果不需要使用SMBv1协议，建议在所有Windows设备上禁用它。禁用SMBv1可以减少暴露在该漏洞下的攻击面。可以通过以下命令禁用：  bash  sc.exe config lanmanworkstation depend= bowser/mrxsmb20/nsi  sc.exe config mrxsmb10 start= disabled  **3、加强网络安全措施**:  **启用防火墙**: 确保启用了防火墙，并配置规则来阻止445端口（SMB协议使用的端口）上的不必要的入站和出站流量。  **隔离关键系统**: 对企业网络进行分段和隔离，防止关键系统与潜在的感染源直接通信。  **4、监控和检测**: 使用入侵检测系统（IDS）或入侵防御系统（IPS）来监控网络流量，识别并阻止试图利用MS17-010漏洞的攻击。 |

* + - 1. 测试输入

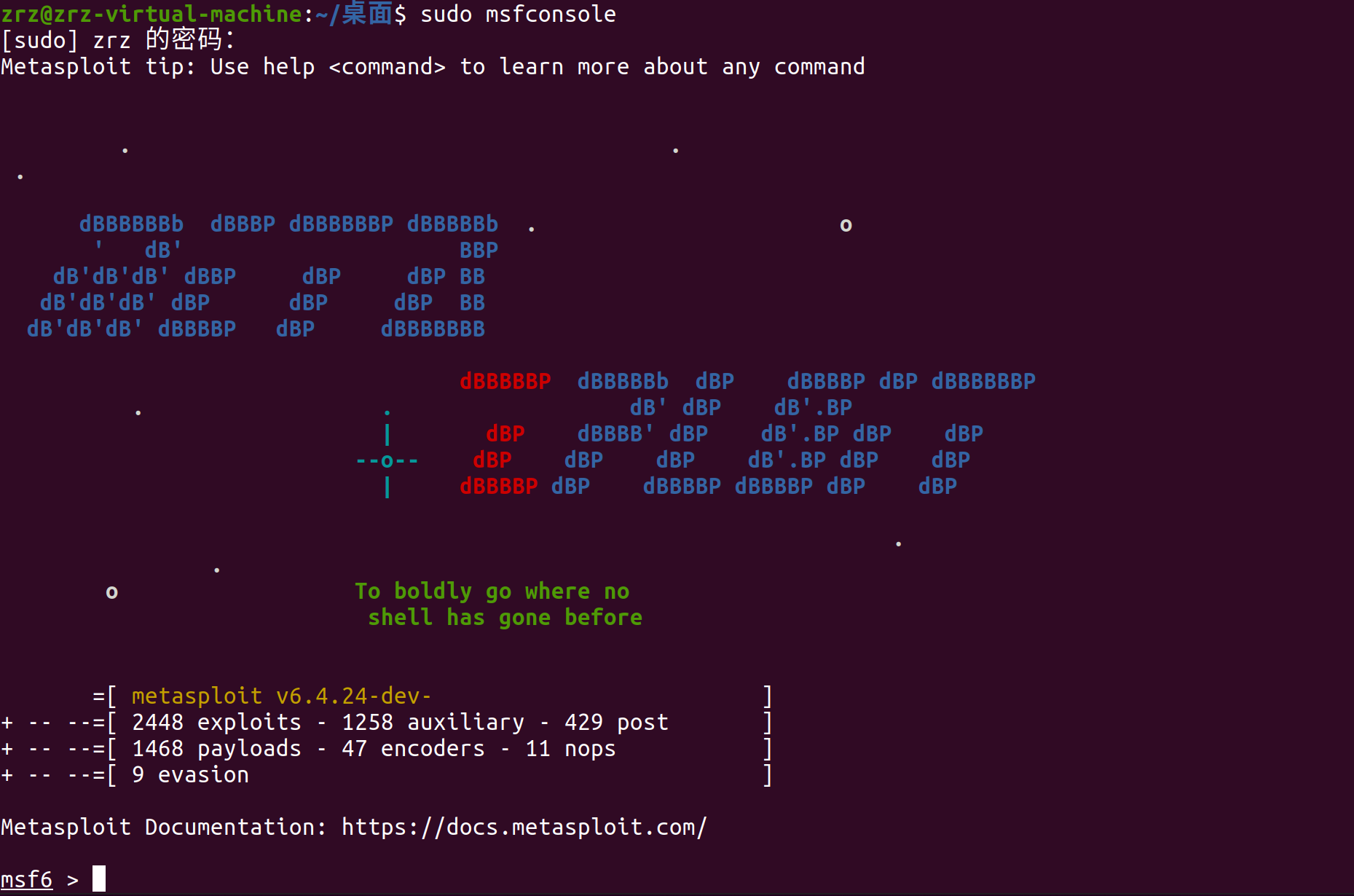
表 3 测试输入字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 目标IP | ip | 192.168.75.132 |
| 目标端口 | port | 445 |
| 请求路径 | path | exploit/windows/smb/ms17\_010\_psexec |

* + - 1. 测试过程

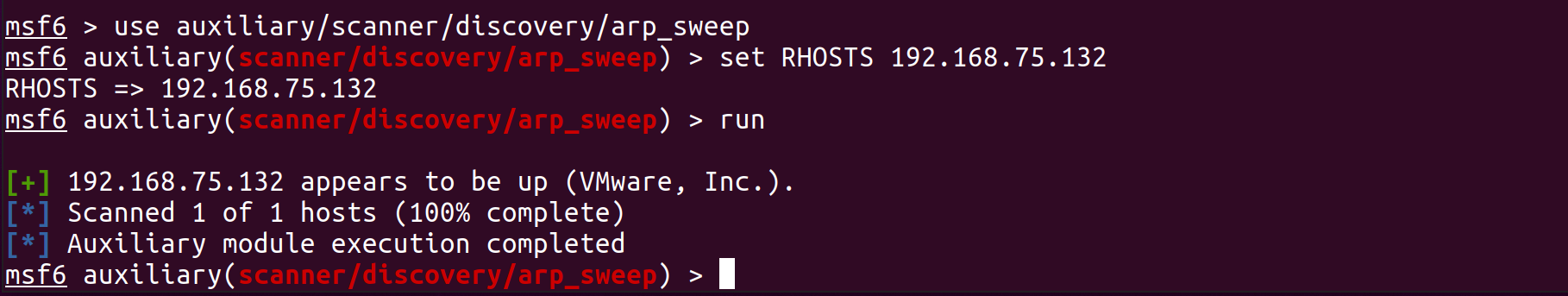
开启MSF:

由于执行ARP探测时使用到root权限，因此以超级用户启动msfconsole:



ARP存活探测：

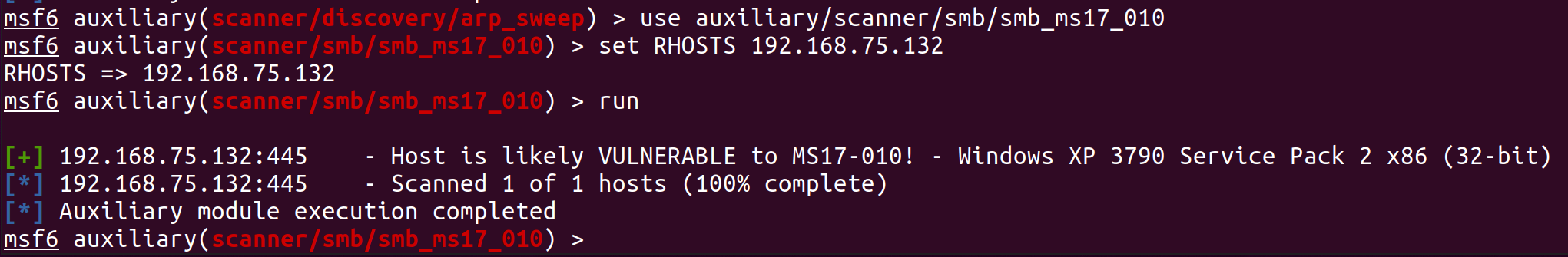
首先使用 arp\_sweep 模块进行 ARP 存活探测，以确定在局域网中存活的主机：



主机存活，以便进行下面的测试。

SMB漏洞扫描测试：

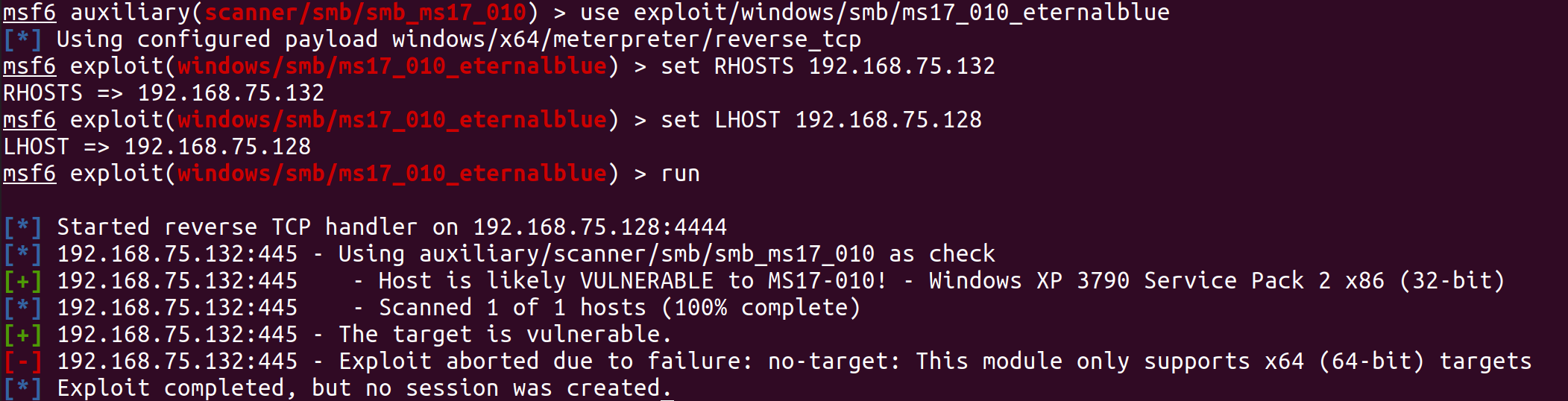
则进一步使用 smb\_ms17\_010 模块进行 SMB 漏洞扫描，以识别目标主机是否受 MS17-010 漏洞的影响：



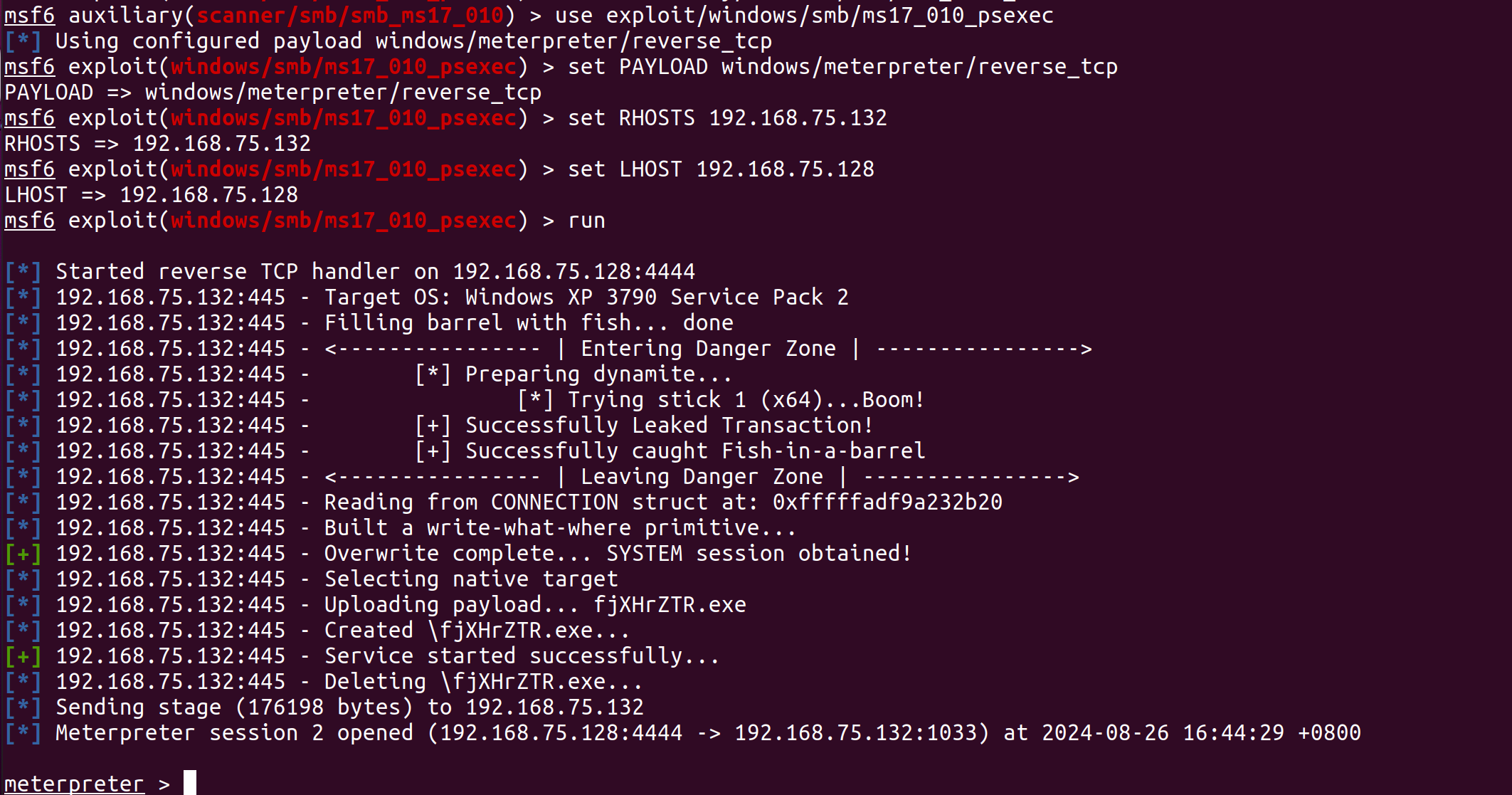
可发现该主机极有可能受到MS17-010漏洞的影响。

Eternalblue载荷攻击：

使用 exploit/windows/smb/ms17\_010\_eternalblue 模块进行 SMB 漏洞利用：



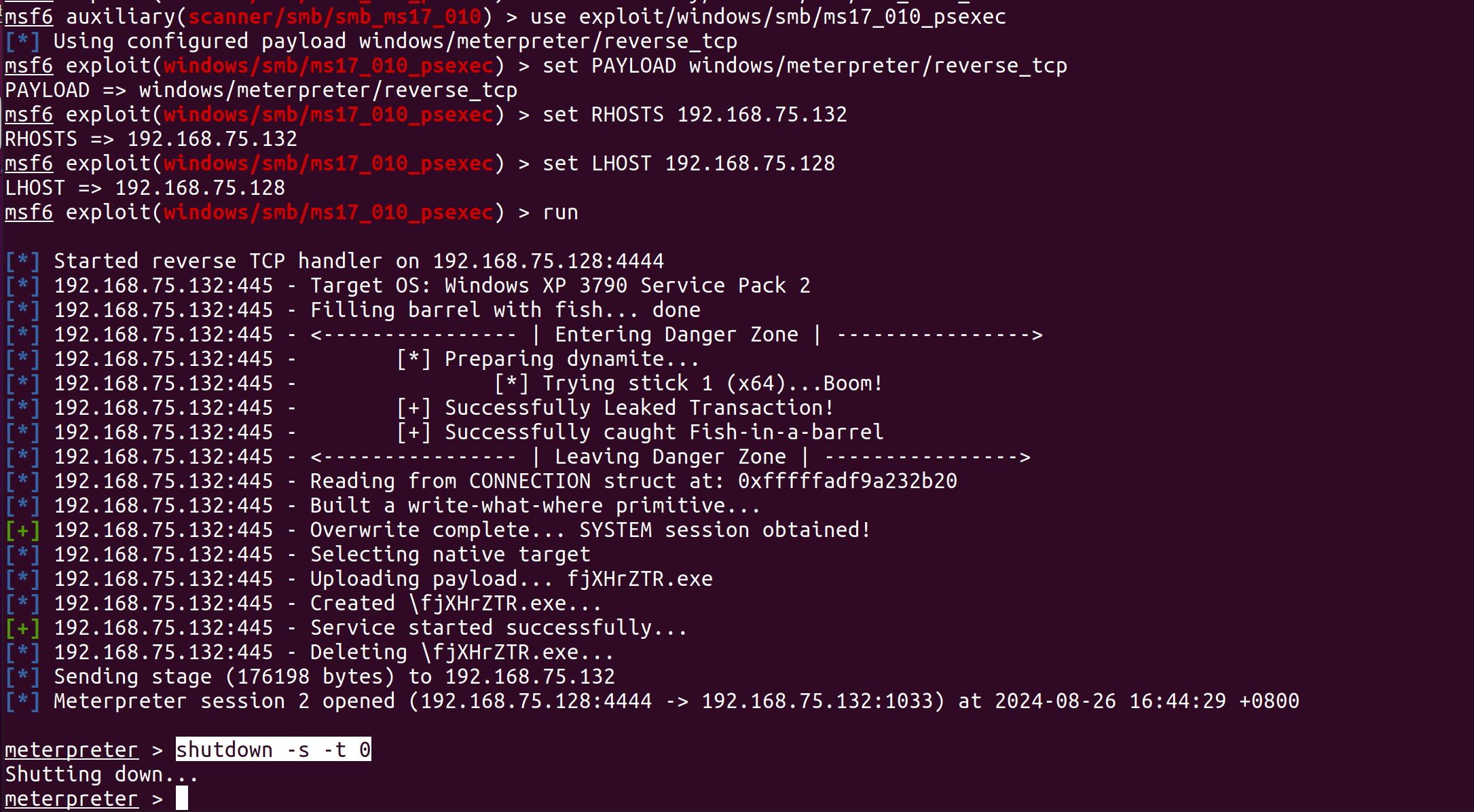
由图示可知，该XP系统使用服务包为32bit，因此使用如下模块进行攻击(注：可使用show payloads命令查看支持的payloads)：



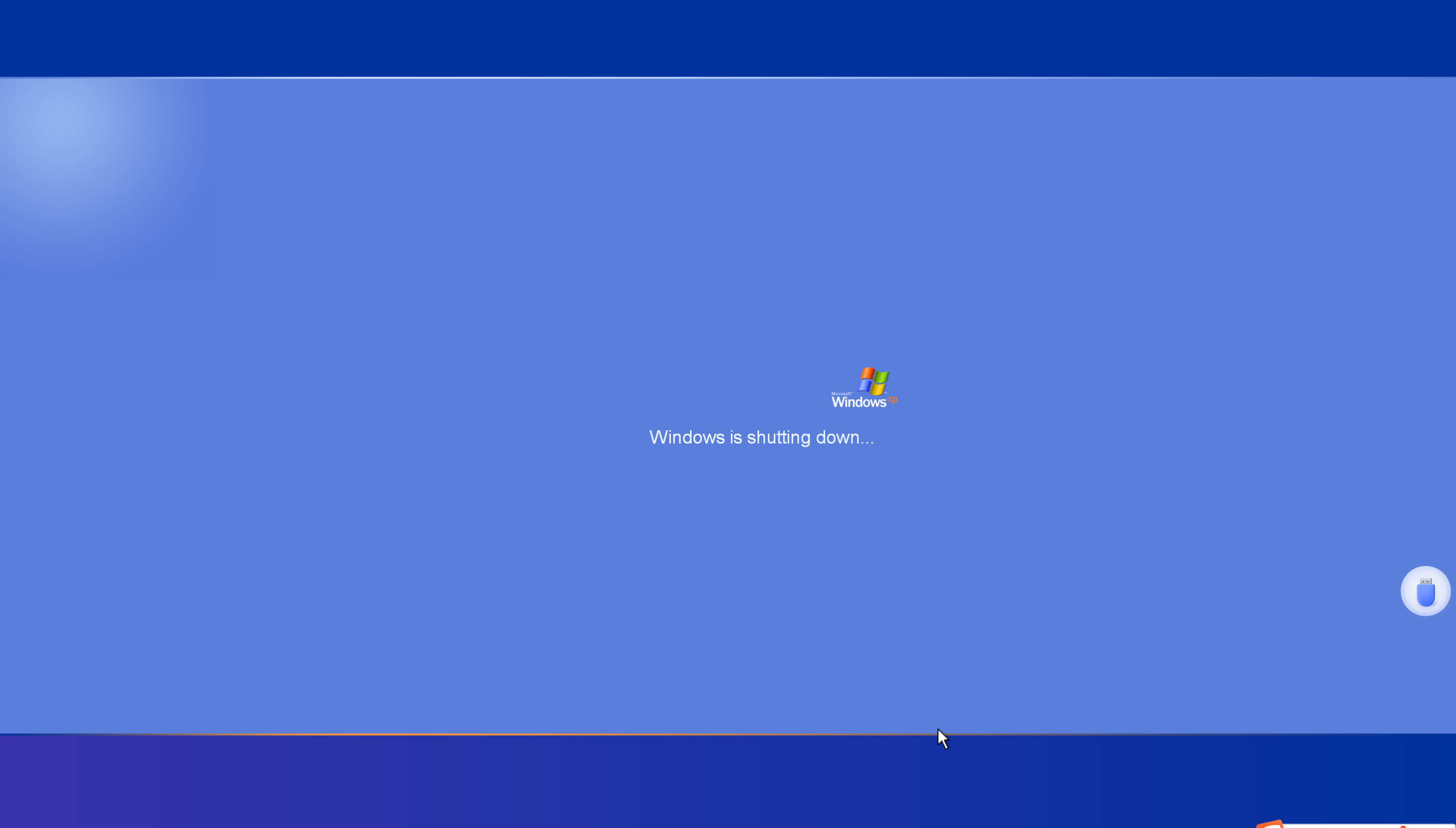
可见获得shell权限。

测试shell权限有效性：

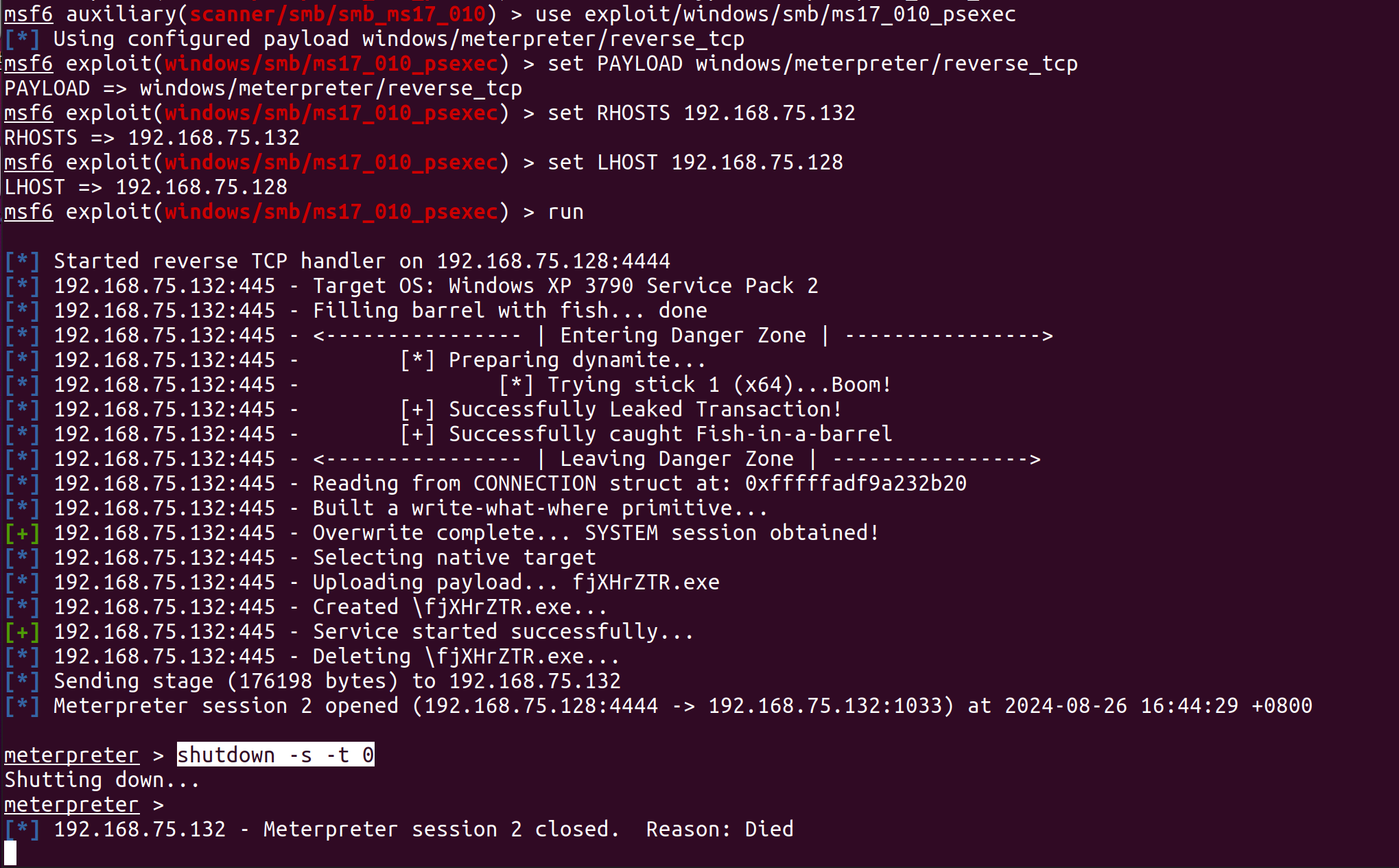
使用`shutdown -s -t 0`命令关闭靶机，测试shell是否有效：



可见winxp关机：



攻击机Ubuntu收到回显，shell关闭：



* + - 1. 测试结果

表 4 测试结果字段说明

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段标识 | 字段描述 |
| 验证结果 | result | True |
| 目标信息 | target\_info | IP：192.168.75.132  OpenPORT：445  SystemVersion：WindowsXP 3790 Service Pack 2 x86 (32 bit) |
| 附加信息 | additional | ReverseTCP端口：4444（句柄）->1033（靶机） |

1. （规范性）

参考文献

1. GB/T 20985.2—2020　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. GB/T 22239—2019　信息安全技术　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

