

Kali linux渗透测试

苑房弘 FANGHONG.YUAN@163.COM




第七章 主动信息收集



主动信息收集

- 直接与目标系统交互通信
- 无法避免留下访问的痕迹
- 使用受控的第三方电脑进行探测
 - 使用代理或已经被控制的主机
 - 做好被封杀的准本
 - 使用噪声迷惑目标，淹没真实的探测流量
- 扫描
 - 发送不同的探测，根据返回结果判断目标状态

发现

- 识别活着的主机
 - 潜在的被攻击目标
 - 输出一个IP地址列表
 - 2、3、4层发现
- 
- A series of white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

发现——二层发现

- 优点：扫描速度快、可靠
- 缺点：不可路由
- Arp协议
 - 抓包

OSI model	Layer description	Protocols
Layer 7 – Application	This layer involves the application software that is sending and receiving data	HTTP, FTP, and Telnet
Layer 6 – Presentation	This layer defines how data is formatted or organized	ASCII, JPEG, PDF, PNG, and DOCX
Layer 5 – Session	This layer involves application session control, management, synchronization, and termination	NetBIOS, PPTP, RPC, and SOCKS
Layer 4 – Transport	This layer involves end-to-end communication services	TCP and UDP
Layer 3 – Network	This layer involves logical system addressing	IPv4, IPv6, ICMP, and IPSec
Layer 2 – Data link	This layer involves physical system addressing	ARP
Layer 1 – Physical	This layer involves the data stream that is passed over the wire	

发现——二层发现

- arping
- arping 1.1.1.1 -c 1
- arping 1.1.1.1 -d
- arping -c 1 1.1.1.1 | grep "bytes from" | cut -d" " -f 5 | cut -d "(" -f 2 | cut -d ")" -f 1
- 脚本
 - arping1.sh eth0 > addrs
 - arping2.sh addrs

发现——二层发现

- `nmap 1.1.1.1-254 -sn`
- `nmap -iL iplist.txt -sn`
- Nmap很强大，后面单独介绍


发现——二层发现

- Netdiscover
 - 专用于二层发现
 - 可用于无线和交换网络环境
 - 主动和被动探测
- 主动
 - `netdiscover -i eth0 -r 1.1.1.0/24`
 - `netdiscover -l iplist.txt`
- 被动
 - `netdiscover -p`
 - 主动arp容易触发报警

发现——二层发现

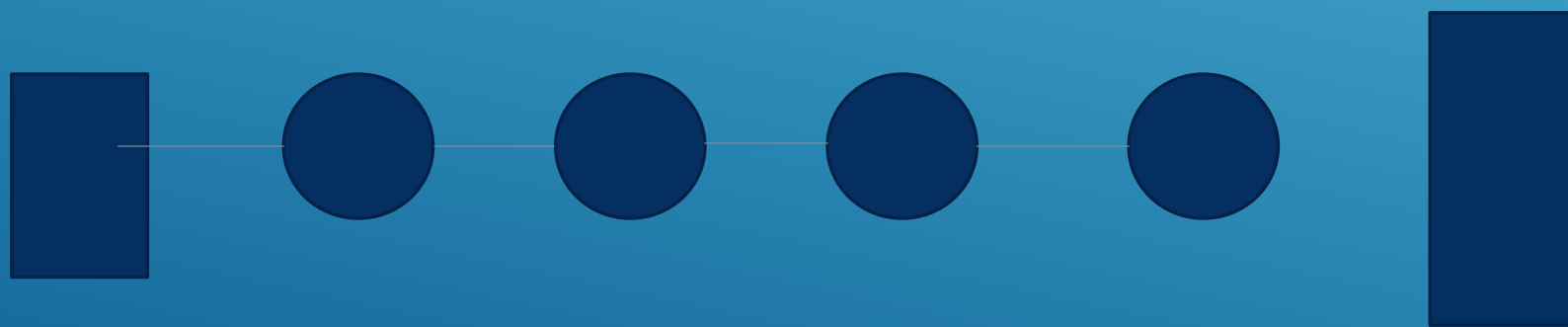
- Scapy
 - 作为Python库进行调用
 - 也可作为单独的工具使用
 - 抓包、分析、创建、修改、注入网络流量
- apt-get install python-gnuplot
- Scapy
 - ARP().display()
 - Sr1()
- Python脚本
 - Arp1.py
 - Arp2.py

发现——三层发现

- 优点
 - 可路由
 - 速度比较快
 - 缺点
 - 速度比二层慢
 - 经常被边界防火墙过滤
 - IP、icmp协议
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, serving as a decorative element.

发现——三层发现

- Ping 1.1.1.1 -c 2
- Ping -R 1.1.1.1 / traceroute 1.1.1.1
- ping 1.1.1.1 -c 1 | grep "bytes from" | cut -d " " -f 4 | cut -d ":" -f 1
- 脚本
 - Pinger.sh 1.1.1.0



发现——三层发现

- Scapy
 - OSI多层堆叠手工声称ICMP包——IP/ICMP
 - ip=IP()
 - ip.dst="1.1.1.1"
 - ping=ICMP()
 - a=srl(ip/ping)
 - a.display()
 - Ping不存在的地址
 - a=srl(ip/ping,timeout=1)
- a = srl(IP(dst="1.1.1.1")/ICMP(),timeout=1)


发现——三层发现

- `pinger1.py 1.1.1.0 > addr`
- `pinger2.py addr`

发现——三层发现

- `nmap -sn 1.1.1.1-255`
- `nmap -iL iplist.txt -sn`

发现——三层发现

- `fping 1.1.1.1 -c 1`
 - `fping -g 1.1.1.1 1.1.1.2`
 - `fping -g 1.1.1.0/24`
 - `fping -f iplist.txt`
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

发现——三层发现

- Hping
 - 能够发送几乎任意TCP/IP包
 - 功能强大但每次只能扫描一个目标 😞
- `hping3 1.1.1.1 --icmp -c 2`
- `for addr in $(seq 1 254); do hping3 1.1.1.$addr --icmp -c 1 >> handle.txt & done`

发现——四层发现

- 优点
 - 可路由且结果可靠
 - 不太可能被防火墙过滤
 - 甚至可以发现所有端口都被过滤的主机
- 缺点
 - 基于状态过滤的防火墙可能过滤扫描
 - 全端口扫描速度慢
- TCP
 - 未经请求的ACK——RST
 - SYN——SYN/ACK、RST
- UDP
 - ICMP端口不可达、一去不复返


发现——四层发现

- ACK——TCP Port——RST
- Scapy
 - `i = IP()`
 - `i.dst="1.1.1.1"`
 - `t = TCP()`
 - `t.flags='A'`
 - `r = (i/t)`
 - `a = sr1(r)`
 - `a.display()`
- `a = sr1(IP(dst="1.1.1.1")/TCP(dport=80,flags='A') ,timeout=1))`
- ACK_Ping.py

发现——四层发现

- UDP——UDP Port——ICMP
- `u = UDP()`
- `u.dport= 33333`
- `r = (i/u)`
- `a = sr1(r,timeout=1,verbose=1)`
- `A.display()`
 - ICMP
- `UDP_Ping.py`
 - UDP发现不可靠


发现——四层发现

- `nmap 1.1.1.1-254 -PU53 -sn`
 - `nmap 1.1.1.1-254 -PA80 -sn`
 - `nmap -iL iplist.txt -PA80 -sn`
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

发现——四层发现

- `hping3 --udp 1.1.1.1 -c 1`
- `for addr in $(seq 1 254); do hping3 -udp 1.1.1.$addr -c 1 >> r.txt; done`
 - `grep Unreachable r.txt | cut -d " " -f 5 | cut -d "=" -f 2`
 - `./udp_hping.sh 1.1.1.0`
- `hping3 1.1.1.1 -c 1` (TCP)
 - `Hping3 1.1.1.1`
 - `./TCP_hping.sh`
 - Flag 0 —— ACK、RST

端口扫描

- 端口对应网络服务及应用端程序
 - 服务端程序的漏洞通过端口攻入
 - 发现开放的端口
 - 更具体的攻击面
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

端口扫描

- UDP端口扫描
 - 假设 ICMP port-unreachable 响应代表端口关闭
 - 目标系统不响应ICMP port-unreachable时，可能产生误判
 - 完整的UPD应用层请求
 - 准确性高
 - 耗时巨大

端口扫描

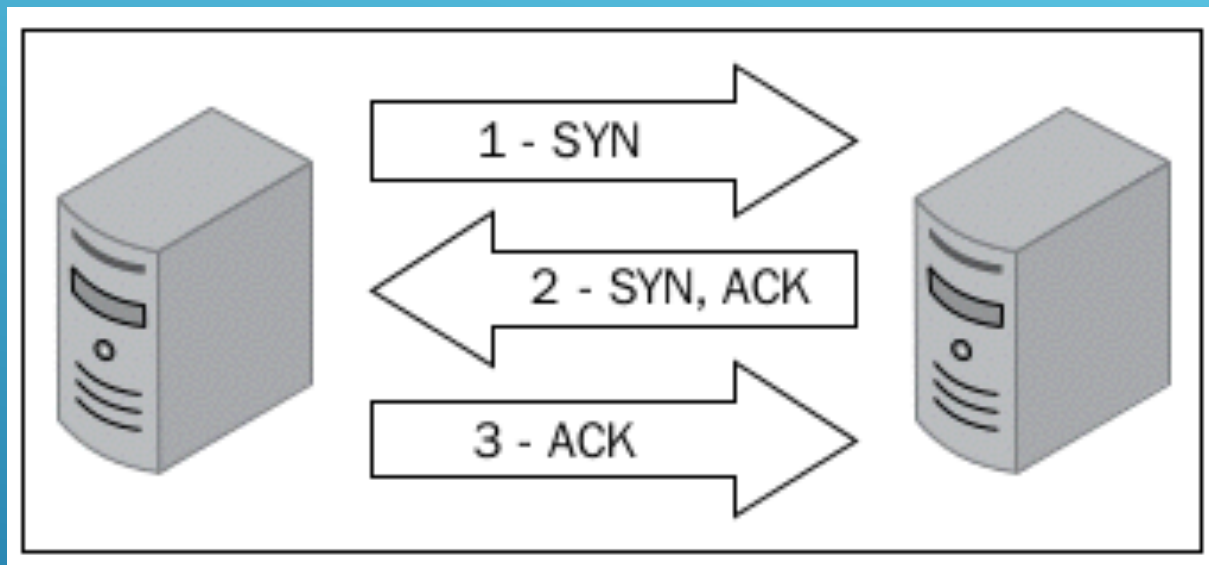
- Scapy UDP Scan
 - 端口关闭: ICMP port-unreachable
 - 端口开放: 没有回包
 - 了解每一种基于UDP的应用层包结构很有帮助
 - 与三层相同的技术
 - 误判
- Scapy
 - `sr1(IP(dst="1.1.1.1")/UDP(dport=53),timeout=1,verbose=1)`
- `./udp_scan.py 1.1.1.1 1 100`

端口扫描

- Nmap
- `nmap -sU 1.1.1.1`
 - 默认的1000个参数
 - ICMP host-unreachable
- `nmap 1.1.1.1 -sU -p 53`
- `nmap -iL iplist.txt -sU -p 1-200`

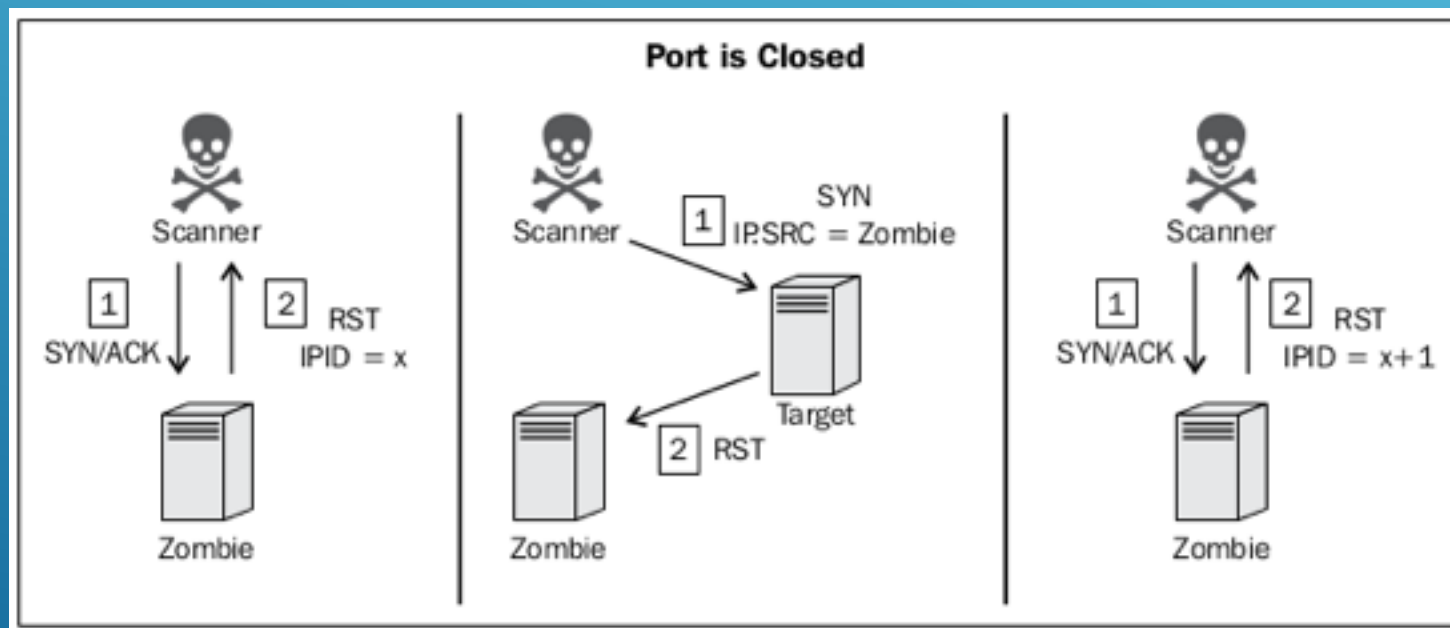
端口扫描

- TCP端口扫描
 - 基于连接的协议
 - 三次握手
 - 隐蔽扫描
 - 僵尸扫描
 - 全连接扫描
 - 所有的TCP扫描方式都是基于三次握手的变化来判断目标端口状态



端口扫描


- 隐蔽扫描——syn
 - 不建立完整连接
 - 应用日志不记录扫描行为——隐蔽
- 僵尸扫描
 - 极度隐蔽
 - 实施条件苛刻
 - 可伪造源地址
 - 选择僵尸机
 - 闲置系统
 - 系统使用递增的IPID
 - 0
 - 随机



隐蔽端口扫描

- Syn——syn/ack——rst
- Scapy
 - `sr1(IP(dst="192.168.60.3")/TCP(dport=80),timeout=1,verbose=1)`
 - `./syn_scan.py`

隐蔽端口扫描

- `nmap -sS 1.1.1.1 -p 80,21,25,110,443`
 - `nmap -sS 1.1.1.1 -p 1-65535 --open`
 - `nmap -sS 1.1.1.1 -p- --open`
 - `nmap -sS -iL iplist.txt -p 80,21,22,23`
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.


隐蔽端口扫描

- hping3
- hping3 1.1.1.1 --scan 80 -S
- hping3 1.1.1.1 --scan 80,21,25,443 -S
- hping3 1.1.1.1 --scan 0-65535 -S
- hping3 -c 10 -S --spooof 1.1.1.2 -p ++1 1.1.1.3


全连接端口扫描

- Scapy
 - Syn扫描不需要raw packets
 - 内核认为syn/ack是非法包，直接发rst终断连接
 - 全连接扫描对scapy比较困难
- `sr1(IP(dst="192.168.20.2")/TCP(dport=22,flags='S'))`
- `./tcp_scan1.py`
- `./tcp_scan2.py`
- `iptables -A OUTPUT -p tcp --tcp-flags RST RST -d 192.168.20.2 -j DROP`

全连接端口扫描

- `nmap -sT 1.1.1.1 -p 80`
 - `nmap -sT 1.1.1.1 -p 80,21,25`
 - `nmap -sT 1.1.1.1 -p 80-2000`
 - `nmap -sT -iL iplist.txt -p 80`
 - 默认1000个常用端口
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, serving as a decorative element.

全连接端口扫描

- dmitry
 - 功能简单，但使用简便
 - 默认150个最常用的端口
 - `dmitry -p 172.16.36.135`
 - `dmitry -p 172.16.36.135 -o output`
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

全连接端口扫描

- `nc -nv -w 1 -z 192.168.60.4 1-100`
- `for x in $(seq 20 30); do nc -nv -w 1 -z 1.1.1.1 $x; done | grep open`
- `for x in $(seq 1 254); do nc -nv -w 1 -z 1.1.1.$x 80; done`

僵尸扫描


- Scapy —— zombie.py

- i=IP()
- t=TCP()
- rz=(i/t)
- rt=(i/t)
- rz[IP].dst=IPz
- rz[TCP].dport=445
- rt[IP].src=IPz
- rt[IP].dst=IPt
- rt[TCP].dport=22
- az1=sr1(rz) / at=sr1(rt) / az2=sr1(rz)
- az1.display() / az2.display()

僵尸扫描

- 发现僵尸机
 - `nmap -p445 192.168.1.133 --script=ipidseq.nse`
- 扫描目标
 - `nmap 172.16.36.135 -sl 172.16.36.134 -Pn -p 0-100`

服务扫描

- 识别开放端口上运行的应用
 - 识别目标操作系统
 - 提高攻击效率
 - Banner捕获
 - 服务识别
 - 操作系统识别
 - SNMP分析
 - 防火墙识别
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

服务扫描

- Banner
 - 软件开发商
 - 软件名称
 - 服务类型
 - 版本号
 - 直接发现已知的漏洞和弱点
- 连接建立后直接获取banner
- 另类服务识别方法
 - 特征行为和响应字段
 - 不同的响应可用于识别底层操作系统

服务扫描

- SNMP
 - 简单网络管理协议
 - Community strings
 - 信息查询或重新配置
 - 识别和绕过防火墙筛选
- 
- A series of three parallel white diagonal lines extending from the bottom right corner towards the center of the slide.

服务扫描——BANNER

- nc -nv 1.1.1.1 22

服务扫描——BANNER

- Python socket
 - Socket模块用于连接网络服务
- import socket
- bangrab = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
- bangrab.connect(("1.1.1.1", 21))
- bangrab.recv(4096)
 - '220 (vsFTPd 2.3.4)\r\n'
- bangrab.close()
- exit()
- Banner不允许抓取, recv函数无返回将挂起!!
- ./ban_grab.py 192.168.1.134 1 100

服务扫描——BANNER

- `dmitry -p 172.16.36.135`
- `dmitry -pb 172.16.36.135`

服务扫描——BANNER

- `nmap -sT 1.1.1.1 -p 22 --script=banner`

服务扫描——BANNER

- `amap -B 172.16.36.135 21`
- `amap -B 172.16.36.135 1-65535`
- `amap -B 172.16.36.135 1-65535 | grep on`

服务扫描——服务识别

- Banner信息抓取能力有限
- nmap响应特征分析识别服务
 - 发送系列复杂的探测
 - 依据响应特征signature
- `nc -nv 1.1.1.1 80`
- `nmap 1.1.1.1 -p 80 -sV`

服务扫描——服务识别

- Amap
- amap 192.168.1.134 80
- amap 172.16.36.135 20-30
- amap 172.16.36.135 20-30 -q
- amap 172.16.36.135 20-30 -qb

操作系统识别

- 操作系统识别技术
 - 种类繁多
 - 好产品采用多种技术组合
- TTL起始值
 - Windows : 128 (65——128)
 - Linux / Unix : 64 (1-64)
 - 某些 Unix : 255

操作系统识别

- python
 - from scapy.all import *
 - win="1.1.1.1"
 - linu="1.1.1.2"
 - aw=sr1(IP(dst=win)/ICMP())
 - al=sr1(IP(dst=linu)/ICMP())
 - if al[IP].ttl<=64:
 - print "host is Linux"
 - else:
 - print "host is windows"
- ./ttl_os.py

操作系统识别

- nmap 使用多种技术识别操作系统
 - nmap 1.1.1.1 -O
 - 系统服务特征

操作系统识别

- xprobe2 1.1.1.1
- 结果有误差

操作系统识别

- 被动操作系统识别
 - IDS
 - 抓包分析
- 被动扫描
- p0f
 - 结合ARP地址欺骗识别全网OS

SNMP扫描

- snmp
 - 信息的金矿
 - 经常被错误配置
 - public / private / manager
- MIB Tree
 - SNMP Management Information Base (MIB)
 - 树形的网络设备管理功能数据库
 - 1.3.6.1.4.1.77.1.2.25
- onesixtyone 1.1.1.1 public
- onesixtyone -c dict.txt -i hosts -o my.log -w 100

SNMP扫描

- `snmpwalk 192.168.20.199 -c public -v 2c`
- 用户
 - `snmpwalk -c public -v 2c 1.1.1.1 1.3.6.1.4.1.77.1.2.25`
- `snmpcheck -t 192.168.20.199`
- `snmpcheck -t 192.168.20.199 -c private -v 2`
- `snmpcheck -t 192.168.20.199 -w`

SMB扫描

- Server Message Block 协议
 - 微软历史上出现安全问题最多的协议
 - 实现复杂
 - 默认开放
 - 文件共享
 - 空会话未身份认证访问 (SMB1)
 - 密码策略
 - 用户名
 - 组名
 - 机器名
 - 用户、组SID

版本	操作系统
SMB1	Windows 2000 / XP / Windows 2003
SMB2	Windows Vista SP1 / Windows 2008
SMB2.1	Windows 7 / Windows 2008 R2
SMB3	Windows 8 / Windows 2012

SMB扫描

- `nmap -v -p139,445 192.168.60.1-20`
- `nmap 192.168.60.4 -p139,445 --script=smb-os-discovery.nse`
- `nmap -v -p139,445 --script=smb-check-vulns --script-args=unsafe=1 1.1.1.1`
- `nbtscan -r 192.168.60.0/24`
- `enum4linux -a 192.168.60.10`

SMTP扫描

- `nc -nv 1.1.1.1 25`
 - `VRFY root`
- `nmap smtp.163.com -p25 --script=smtp-enum-users.nse --script-args=smtp-enum-users.methods={VRFY}`
- `nmap smtp.163.com -p25 --script=smtp-open-relay.nse`
- `smtp-user-enum -M VRFY -U users.txt -t 10.0.0.1`
- `./smtp.py`

防火墙识别

- 通过检查回包，可能识别端口是否经过防火墙过滤
- 设备多种多样，结果存在一定误差

	Send	Response	Type
1	SYN	No	Filtered
	ACK	RST	
2	SYN	SYN+ACK / SYN+RST	Filtered
	ACK	No	
3	SYN	SYN+ACK / SYN+RST	Unfiltered / Open
	ACK	RST	
4	SYN	No	Closed
	ACK	No	

防火墙识别

- scapy
- python
 - ./fw_detect.py 1.1.1.1 443

防火墙识别

- Nmap有系列防火墙过滤检测功能
- `nmap -sA 172.16.36.135 -p 22`

负载均衡识别

- 广域网负载均衡
 - DNS
 - HTTP-Loadbalancing
 - Nginx
 - Apache
 - lbd www.baidu.com
 - lbd mail.163.com
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract design element.

WAF识别

- WEB应用防火墙
 - wafw00f -I
 - wafw00f http://www.microsoft.com
 - nmap www.microsoft.com --script=http-waf-detect.nse
- 
- A series of white diagonal lines of varying lengths and thicknesses, located in the bottom right corner of the slide, creating a modern, abstract design element.

NMAP

- 所有参数
- zenmap

Q & A

