

详解常见漏洞扫描器及网络扫描技术

随着互联网的飞速发展,网络安全逐渐成为一个潜在的巨大问题,如何保障自身网络的安全,其中一个主要的方法就是自查自纠,而在这个过程中,对自己的网络进行扫描成为一种较为快捷、直观、简单的方法。扫描技术基于 TCP/IP 协议,对各种网络服务,无论是主机或者防火墙、路由器都适用。同时,扫描可以确认各种配置的正确性,避免遭受不必要的攻击。网络扫描是一把双刃剑,对于安全管理员来说,可以用来确保自己系统的安全性,也能被黑客用来查找系统的入侵点。目前,扫描的技术已经非常成熟,已经有大量的商业、非商业的扫描器在各组织里使用。

扫描器的重要性体现在:

扫描器能够暴露网络上潜在的脆弱性。

无论扫描器被管理员利用,或者被黑客利用,都有助于加强系统的安全性。

它能使得漏洞被及早发现,而漏洞迟早会被发现的。

扫描器可以满足很多人的好奇心。

扫描器除了能扫描端口,往往还能够:发现系统存活情况,以及哪些服务在运行;

用已知的漏洞测试这些系统;

对一批机器进行测试,简单的迭代过程;

有进一步的功能,包括操作系统辨识、应用系统识别。

1、常见漏洞扫描器类型

1.1 端口扫描器

Nmap 被称为扫描器之王。这种类型扫描器很容易引起误解,很多人认为

对于 nmap 这种 port scanner，功能仅限于扫描端口，这实际上是一个很大的误区，如果我们看一下它的功能，就会发现功能还是非常广泛的，包括操作系统的服务判定、操作系统指纹的判定、防火墙及 IDS 的规避技术。Nmap 从技术角度来说是非常出色的，并且可以完成大范围的早期评估工作。实际上 nmap 的端口扫描的不管是主机开放端口、服务、操作系统版本，它的大部分依据都来自于端口扫描的结果，根据其结果去判定其他信息。所以，认为 nmap 只能扫描端口是一个误区。

1.2 漏洞扫描器

以 nessus 为免费产品代表，nessus 的安装应用程序、脚本语言都是公开的，但从版本 3 开始它就转向一个私有的授权协议，其扫描引擎仍然免费，不过对其支持和最新的漏洞定义要收费，不过收费是有时间限制的，如果不想付钱的话，可以等待 15 天，15 天之后，其大部分插件都将是免费插件。Nessus 目前最新版本到了 3.2。这种扫描器不仅可以检查系统漏洞，还可以检查一部分的配置失误。

1.3 WEB 应用扫描器

这类扫描器相对而言，做的比较专，仅用于评估网站的安全性，对于系统、网络的基础情况一般不关注，关注的焦点主要是 WEB 应用。常见的有 appscan、webinspect。主要检测 WEB 应用数据提交、信息泄露等问题。

2、商业扫描器特点

基本上大部分商业扫描器都工作在黑盒模式，在这种模式下无法看到源代码，以一个近似于渗透者或攻击者的身份去看待需要评估的系统。这种扫描器特点有：

2.1 漏洞精确扫描

由于在商业化应用中，对误报、漏报的容忍程度比较低。但目前的情况，误报和漏报还是无法规避的。具体扫描的信息有：

状态扫描：即其开放的服务、通信的情况、OS 版本、应用服务的版本。

漏洞扫描（验证）：验证当前系统是否存在可以利用、不可以利用的漏洞，如果可以利用，某些扫描器可以进行写入文件或者拿到 shell 之类的功能。

弱口令扫描：对于开放的服务进行弱口令扫描，这也是很重要的一个功能。

2.2 修补措施

商用扫描器在漏洞精确扫描之后，会给出一些建议和技术手段来屏蔽漏洞。最初是提供一些修补建议，这种方式对专业技术人员来说有相当价值，但对于一些技术较薄弱或者比较懒惰的用户，修补建议的作用就被忽略了。在新一代的商用扫描器中，提出了修补联动的概念，通过发送注册表去提示用户，用户双击注册表，就可以导入需要修改、升级补丁的信息，并且还可以和 WSUS 进行联动。这样就可以基本上达到自动化的修补。

3、常见扫描主要技术

要了解常见扫描的主要技术，还要以扫描器工作的流程开始，以 nmap 为例，整个扫描流程如下：

1、存活性扫描：是指大规模去评估一个较大网络的存活状态。例如跨地域、跨系统的大型企业。但是被扫描主机可能会有一些欺骗性措施，例如使用防火墙阻塞 ICMP 数据包，可能会逃过存活性扫描的判定。

2、端口扫描：针对主机判断端口开放和关闭情况，不管其是不是存活。端口扫描也成为存活性扫描的一个有益补充，如果主机存活，必然要提供相应的状态，因此无法隐藏其存活情况。

3、服务识别：通过端口扫描的结果，可以判断出主机提供的服务及其版本。

4、操作系统识别：利用服务的识别，可以判断出操作系统的类型及其版本。

这就是以 nmap 为代表的扫描过程。

3.1 主机存活扫描技术

主机扫描的目的是确定在目标网络上的主机是否可达。这是信息收集的初级阶段，其效果直接影响到后续的扫描。Ping 就是最原始的主机存活扫描技术，利用 icmp 的 echo 字段，发出的请求如果收到回应的话代表主机存活。

常用的传统扫描手段有：

1. ICMP Echo 扫描 精度相对较高。通过简单地目标主机发送 ICMP Echo Request 数据包，并等待回复的 ICMP Echo Reply 包，如 Ping。

2. ICMP Sweep 扫描：sweep 这个词的动作很像机枪扫射，icmp 进行扫射式的扫描，就是并发性扫描，使用 ICMP Echo Request 一次探测多个目标主机。通常这种探测包会并行发送，以提高探测效率，适用于大范围的评估。

3. Broadcast ICMP 扫描：广播型 icmp 扫描，利用了一些主机在 icmp 实现上的差异，设置 ICMP 请求包的目标地址为广播地址或网络地址，则可以探测广播域或整个网络范围内的主机，子网内所有存活主机都会给以回应。但这种情况只适合于 UNIX/Linux 系统。

4. Non-Echo ICMP 扫描：在 ICMP 协议中不光只有 ICMP ECHO 的 ICMP 查询信息类型，在 ICMP 扫描 技术中也用到 Non-ECHO ICMP 技术（不仅仅能探测主机，也可以探测网络设备如路由）。利用了 ICMP 的服务类型（Timestamp 和 Timestamp Reply、Information Request 和 Information Reply、Address Mask Request 和 Address Mask Reply）。

3.2 规避技术

为到达规避防火墙和入侵检测设备的目的，ICMP 协议提供网络间传送错误信息的功能也成为了主要的扫描手段。其主要原理就是利用被探测主机产生的 ICMP 错误报文来进行复杂的主机探测。

常用的规避技术大致分为 4 类：

1. 异常的 IP 包头：向目标主机发送包头错误的 IP 包，目标主机或过滤设备会反馈 ICMP Parameter Problem Error 信息。常见的伪造错误字段为 Header Length 和 IP Options。不同厂家的路由器和操作系统对这些错误的处理方式不同，返回的结果也不同。

2. 在 IP 头中设置无效的字段值：向目标主机发送的 IP 包中填充错误的字段值，目标主机或过滤设备会反馈 ICMP Destination Unreachable 信息。这种方法同样可以探测目标主机和网络设备。

3. 通过超长包探测内部路由器：若构造的数据包长度超过目标系统所在路由器的 PMTU 且设置禁止分片标志，该路由器会反馈 Fragmentation Needed and Don't Fragment Bit was Set 差错报文。

4. 反向映射探测：用于探测被过滤设备或防火墙保护的网络和主机。构造可能的内部 IP 地址列表，并向这些地址发送数据包。当对方路由器接收到这些数据包时，会进行 IP 识别并路由，对不在其服务的范围的 IP 包发送 ICMP Host Unreachable 或 ICMP Time Exceeded 错误报文，没有接收到相应错误报文的 IP 地址可被认为在该网络中。

举例来看：

```

Internet Protocol, Src Addr: 10.1.1.36
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00
  Total Length: 40
  Identification: 0xf78a
  Flags: 0x00
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (0x06)
  Header checksum: 0x2c24 (correct)
  Source: 10.1.1.36 (10.1.1.36)
  Destination: 10.1.1.252 (10.1.1.252)
Transmission Control Protocol, Src Port

```

图 1

当发送一个数据包的时候，数据包头部会显示其封装类型，这里显示的是 tcp 协议。其编号是 06，看到 06 就可以知道这是一个 tcp 报文。如果把报文标识成不可识别的协议，例如：

```

Fragment offset: 0
Time to live: 42
Protocol: Unknown (0xb2)
Header checksum: 0xc904 (cor
Source: 10.1.1.36 (10.1.1.36)

```

图 2

协议字段里显示是 unknown，编号是 b2，这个协议是未知的，那嗅探器也会返回给一个 unknown。就是这样发送一个不可识别的协议给其他主机的时候，对方主机也无法识别。当不可识别的时候：

```

Ethernet II, Src: 00:00:00:00:00:00, Dst: 00:00:00:00:00:00
Internet Protocol, Src Addr: 10.1.1.1, Dst Addr: 10.1.1.252
Internet Control Message Protocol
  Type: 3 (Destination unreachable)
  Code: 2 (Protocol unreachable)

```

图 3

对方认为自己的协议栈上无法识别，第一个动作就是先把数据包丢弃，丢掉报文之后去通知对方，返回了这条 icmp 的信息。类型为 3，目标不可达，其 code 是 2，告诉你目标不可达的原因是协议不可达。当你发送一个错误的数据包，对方不可识别就返回给你一个 icmp 信息，达到了探测对方的目的。

3.3 端口扫描技术

在完成主机存活性判断之后,就应该去判定主机开放信道的状态,端口就是在主机上面开放的信道,0-1024 为知名端口,端口总数是 65535。端口实际上就是从网络层映射到进程的通道。通过这个关系就可以掌握什么样的进程使用了什么样的通信,在这个过程里面,能够通过进程取得的信息,就为查找后门、了解系统状态提供了有力的支撑。常见流行的端口扫描技术通常有:

3.3.1 TCP 扫描:

利用三次握手过程与目标主机建立完整或不完整的 TCP 连接。

TCP connect()扫描: tcp 的报头里,有 6 个连接标记,分别是 urg、ack、psh、rst、syn、fin。通过这些连接标记不同的组合方式,可以获得不同的返回报文。例如,发送一个 syn 置位的报文,如果 syn 置位瞄准的端口是开放的, syn 置位的报文到达的端口开放的时候,他就会返回 syn+ack,代表其能够提供相应的服务。我收到 syn+ack 后,返回给对方一个 ack。这个过程就是著名的三次握手。这种扫描的速度和精度都是令人满意的。

Reverse-ident 扫描 这种技术利用了 Ident 协议(RFC1413),tcp 端口 113。很多主机都会运行的协议,用于鉴别 TCP 连接的用户。

identd 的操作原理是查找特定 TCP/IP 连接并返回拥有此连接的进程的用户名。它也可以返回主机的其他信息。但这种扫描方式只能在 tcp 全连接之后才有效,并且实际上很多主机都会关闭 ident 服务。

Tcp syn 扫描:向目标主机的特定端口发送一个 SYN 包,如果端口没开放就不会返回 syn+ack,这时会给你一个 rst,停止建立连接。由于连接没有完全建立,所以称为半开放扫描。但由于 syn flood 作为一种 ddos 攻击手段被大量

采用,因此很多防火墙都会对 syn 报文进行过滤,所以这种方法并不能总是有用。

其他还有 fin、NULL、Xmas 等扫描方式。

3.3.2 UDP 扫描

由于现在防火墙设备的流行,tcp 端口的管理状态越来越严格,不会轻易开放,并且通信监视严格。为了避免这种监视,达到评估的目的,就出现了秘密扫描。这种扫描方式的特点是利用 UDP 端口关闭时返回的 ICMP 信息,不包含标准的 TCP 三次握手协议的任何部分,隐蔽性好,但这种扫描使用的数据包在通过网络时容易被丢弃从而产生错误的探测信息。

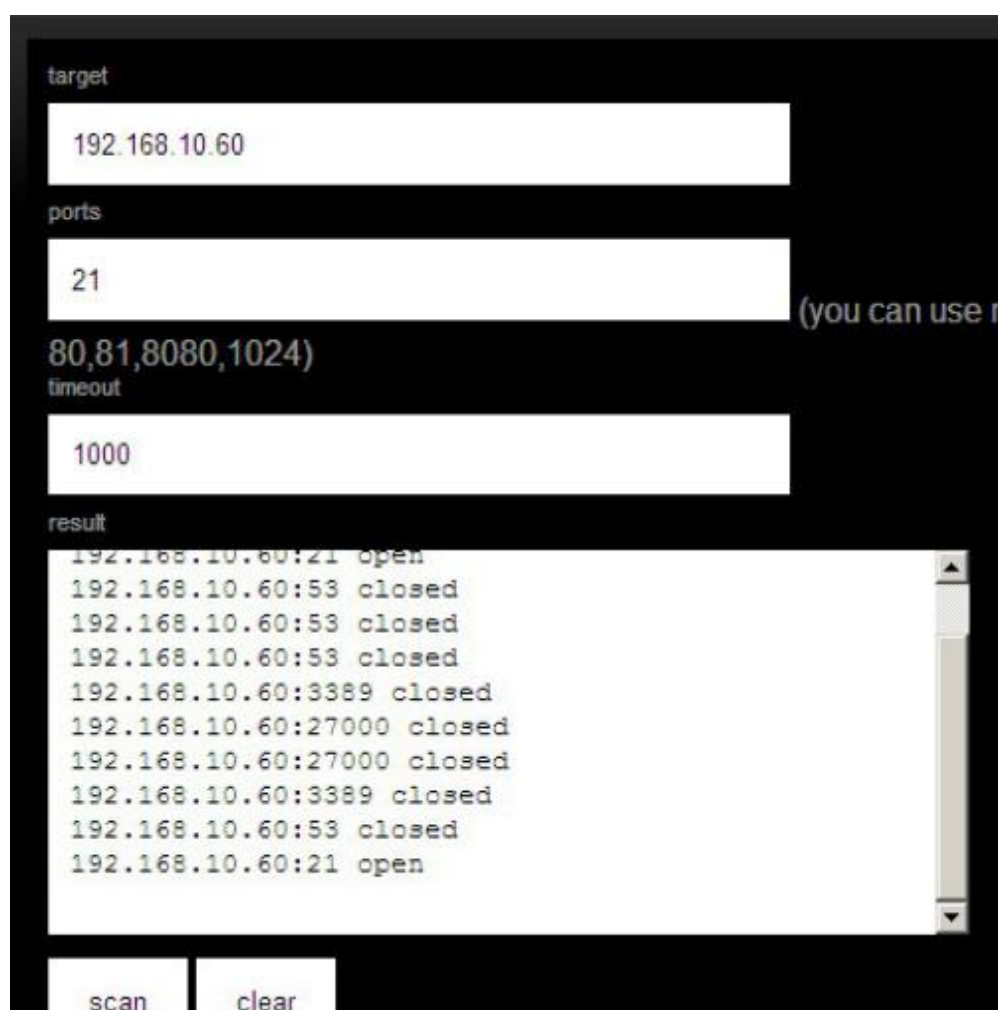
但是,UDP 扫描方式的缺陷很明显,速度慢、精度低。UDP 的扫描方法比较单一,基础原理是:当你发送一个报文给 udp 端口,该端口是关闭状态时,端口会返回给一个 icmp 信息,所有的判定都是基于这个原理。如果关闭的话,什么信息都不发。

Traceroute 扫描 :tracert 向 30000 以上的高端口(一般认为,主机的 30000 以上高端口利用率非常低,任何主机都不会轻易开放这种高端口,默认都是关闭的)。如果对方端口关闭,会返回给 icmp 信息,根据这个往返时间,计算跳数、路径信息,了解延时情况。这是 traceroute 原理,也是从这个原理上演变出来 udp 扫描技术。

使用 udp 扫描要注意的是 1、udp 状态、精度比较差,因为 udp 是不面向连接的,所以整个精度会比较低。2、udp 扫描速度比较慢,tcp 扫描开放 1 秒的延时,在 udp 里可能就需要 2 秒,这是由于不同操作系统在实现 icmp 协议的时候为了避免广播风暴都会有峰值速率的限制(因为 icmp 信息本身并不是传输载荷信息,不会有人拿他去传输一些有价值信息。操作系统在实现的时候是不

希望 icmp 报文过多的。为了避免产生广播风暴，操作系统对 icmp 报文规定了峰值速率，不同操作系统的速率不同）利用 udp 作为扫描的基础协议，就会对精度、延时产生较大影响。

当前在渗透测试过程中对于端口的扫描是非常灵活的，06 年的黑帽大会上，就有人利用了开发了工具探测网内哪台主机打开了 80 端口，这样的技术在当前的互联网上利用的非常普遍。



The image shows a web-based port scanner interface. It has a dark background with white text and input fields. The interface includes sections for 'target', 'ports', 'timeout', and 'result'. The 'target' field contains '192.168.10.60'. The 'ports' field contains '21' and a hint '(you can use n' is visible. The 'timeout' field contains '1000'. The 'result' section shows a list of scan results for the target IP. At the bottom, there are 'scan' and 'clear' buttons.

target
192.168.10.60

ports
21

(you can use n

timeout
1000

result
192.168.10.60:21 open
192.168.10.60:53 closed
192.168.10.60:53 closed
192.168.10.60:53 closed
192.168.10.60:3389 closed
192.168.10.60:27000 closed
192.168.10.60:27000 closed
192.168.10.60:3389 closed
192.168.10.60:53 closed
192.168.10.60:21 open

scan clear

图 4

3.4 服务及系统指纹

在判定完端口情况之后，继而就要判定服务。

3.4.1 根据端口判定

这种判定服务的方式就是根据端口，直接利用端口与服务对应的关系，比如 23 端口对应 telnet，21 对应 ftp，80 对应 http。这种方式判定服务是较早的一种方式，对于大范围评估是有一定价值的，但其精度较低。例如使用 nc 这样的工具在 80 端口上监听，这样扫描时会以为 80 在开放，但实际上 80 并没有提供 http 服务，由于这种关系只是简单对应，并没有去判断端口运行的协议，这就产生了误判，认为只要开放了 80 端口就是开放了 http 协议。但实际并非如此，这就是端口扫描技术在服务判定上的根本缺陷。

3.4.2 BANNER

Banner 的方式相对精确，获取服务的 banner，是一种比较成熟的技术，可以用来判定当前运行的服务，对服务的判定较为准确。而且不仅能判定服务，还能够判定具体的服务版本信息。例如下图，根据头部信息发现对方是 redhat linux，基本上可以锁定服务的真实性。

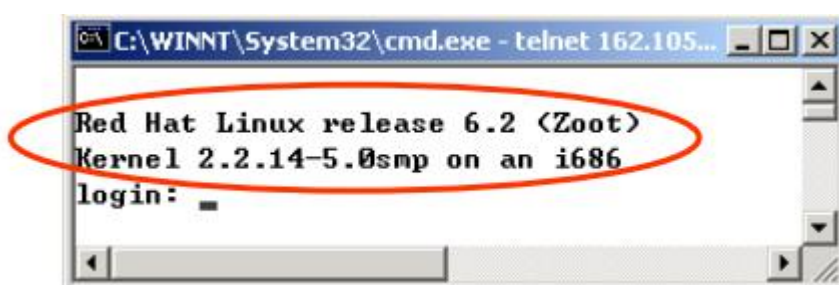


图 5

这种技术比较灵活。像 http,ftp,telnet 都能够获取一些 banner 信息。为了判断服务类型、应用版本、OS 平台，通过模拟各种协议初始化握手，就可以获取信息。

但是在安全意识普遍提升的今天，对 Banner 的伪装导致精度大幅降低。

例如 IIS&Apache：修改存放 Banner 信息的文件字段进行修改，这种修改的开销很低。现在流行的一个伪装工具 Servermask，不仅能够伪造多种主流 Web 服务器 Banner，还能伪造 Http 应答头信息里的项的序列。

使用前

```
$ nc 192.168.7.247 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Location: http://192.168.7.247/Default.htm
Date: Fri, 01 Jan 1999 20:09:05 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Fri, 01 Jan 1999 20:09:05 GMT
ETag: W/"e0d362a4c335be1:ae0"
Content-Length: 133
```

图 6

使用后

```
$ nc 192.168.7.247 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 01 Jan 1999 20:06:24 GMT
Server: Apache/1.3.19 (Unix) (Red-Hat/Linux) mod_ssl/2.8.1
OpenSSL/0.9.6 DAV/1.0.2 PHP/4.0.4pl1 mod_perl/1.24_01
Content-Location: http://192.168.7.247/Default.htm
Last-Modified: Fri, 01 Jan 1999 20:06:24 GMT
ETag: W/"e0d362a4c335be1:ae0"
Accept-Ranges: bytes
Content-Length: 133
Content-Type: text/html
```

图 7

变的不仅是红色的部分，而且整个返回序列都变化了。这个变化是手工难以修改的。如果不能成功的修改序列，那就可能给有经验的渗透者提供依据。

3.4.3 指纹技术

指纹技术利用 TCP/IP 协议栈实现上的特点来辨识一个操作系统。可辨识的 OS 的种类，包括哪些操作系统，甚至小版本号。指纹技术有主动和被动两种。

主动识别技术：采用主动发包，利用多次的试探，去一次一次筛选不同信息，

比如根据 ACK 值判断，有些系统会发送回所确认的 TCP 分组的序列号，有些会发回序列号加 1。还有一些操作系统会使用一些固定的 tcp 窗口。某些操作系统还会设置 IP 头的 DF 位来改善性能。这些都成为判断的依据。这种技术判定 windows 的精度比较差，只能够判定一个大致区间，很难判定出其精确版本，但是在 unix，网络设备时甚至可以判定出小版本号，比较精确。如果目标主机与源主机跳数越多，精度越差。因为数据包里的很多特征值在传输过程中都已经被修改或模糊化，会影响到探测的精度。nmap -O 参数就是其代表。

被动识别技术：不是向目标系统发送分组，而是被动监测网络通信，以确定所用的操作系统。利用对报头内 DF 位，TOS 位，窗口大小，TTL 的嗅探判断。因为并不需要发送数据包，只需要抓取其中的报文，所以叫做被动识别技术。例如 telnet 对方，并用 snort 监听数据包：

```
TCP TTL:255 TOS:0x0 ID:58955 DF
**S***A* Seq:0xD3B709A4 Ack:0xBE09B2B7 Win:0x2798
TCP Options => NOP NOP TS:9688775 9682347 NOP WS:0
MSS:1460
```

得到这些信息后，熟悉系统的人可猜测到操作系统是 Solaris 2.6-2.7。在 nmap 中，也有操作系统的指纹库，可以从指纹库中去匹配。其代表扫描工具有 Siphon，天眼。

ICMP 指纹识别技术：这种工具的出现较晚，大概在 2001-2002 年，在黑客大会上提出，并开发出相应的工具 xprobe，其优势是只需要通过 icmp，发送一批 UDP 包给高端关闭的端口，然后计算返回来的不可达错误消息。通常情

况下送回 IP 头+8 个字节,但是个别系统送回的数据更多一些。根据 ICMP 回应的 TOS、TTL 值、校验和等信息,通过这些信息以树状的形式去过滤,最终精确锁定。下图只是整个判定过程中的一小量部分,类似的图还有很多张。

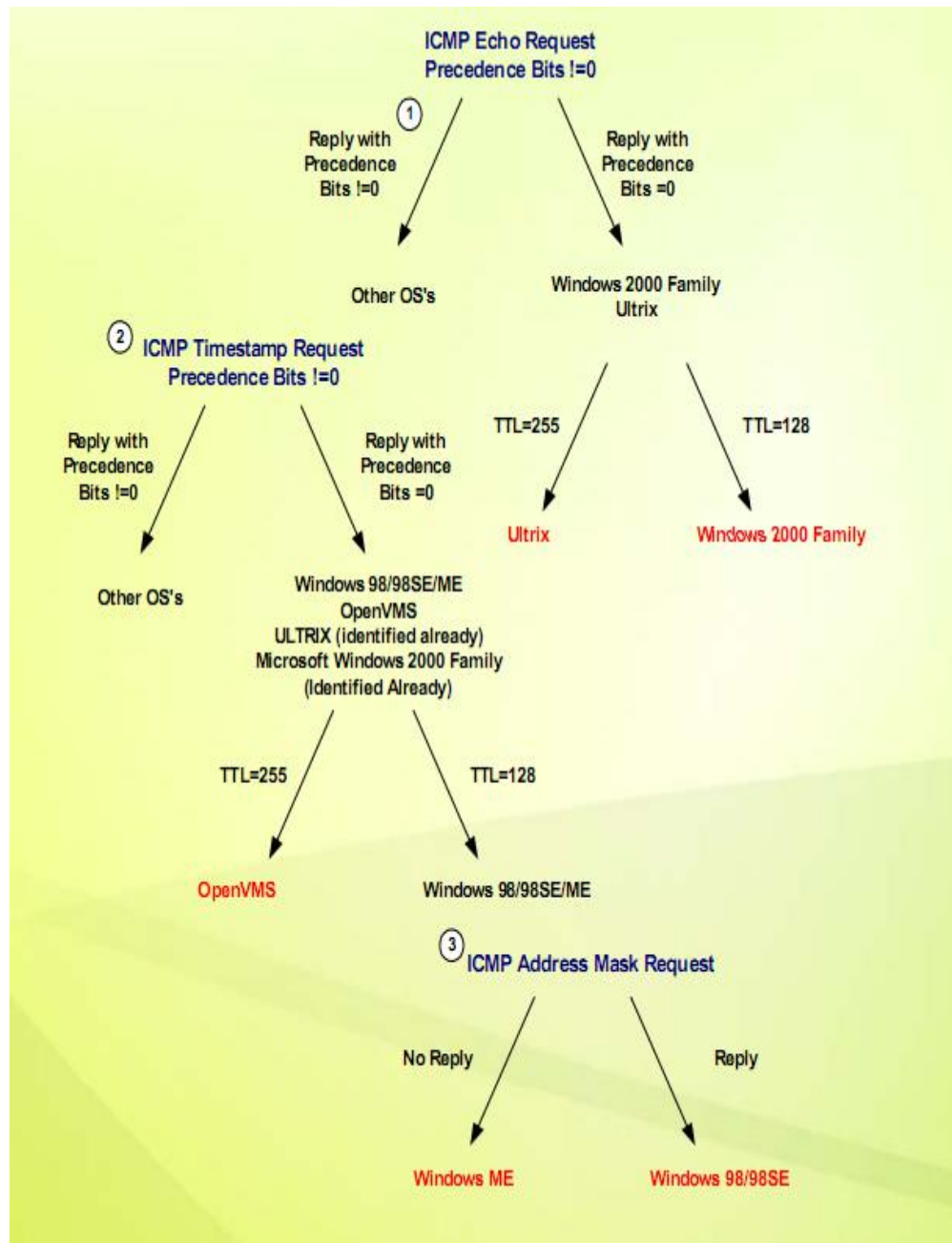


图 8

例如 TTL 值,当你发送 icmp 的 echo 请求,对方回应的 ttl 值是有一定规律可循的,一般是 4 个数值,32 (win95 ,唯一值为 32 的操作系统), 64 , 128

(windows 家族 , 只有 win95 例外) , 256。 64 和 256 比较难以分辨 , 大多数的 unix 和 linux 都可能是。 代表性扫描工具 Xprobe。