

UDP Scanning

John Kristoff

jtk@depaul.edu

+1 312 362-5878

DePaul University
Chicago, IL 60604

What are we talking about?

- Remotely probing hosts using UDP messages
- Comparing UDP, ICMP and TCP scanning
- UDP scanning details
- UDP scanning failure scenarios
- How to make UDP scanning more reliable
- Why is this talk important?
 - A colleague expressed the need for public info
 - But really... to help justify my trip to Hawaii!

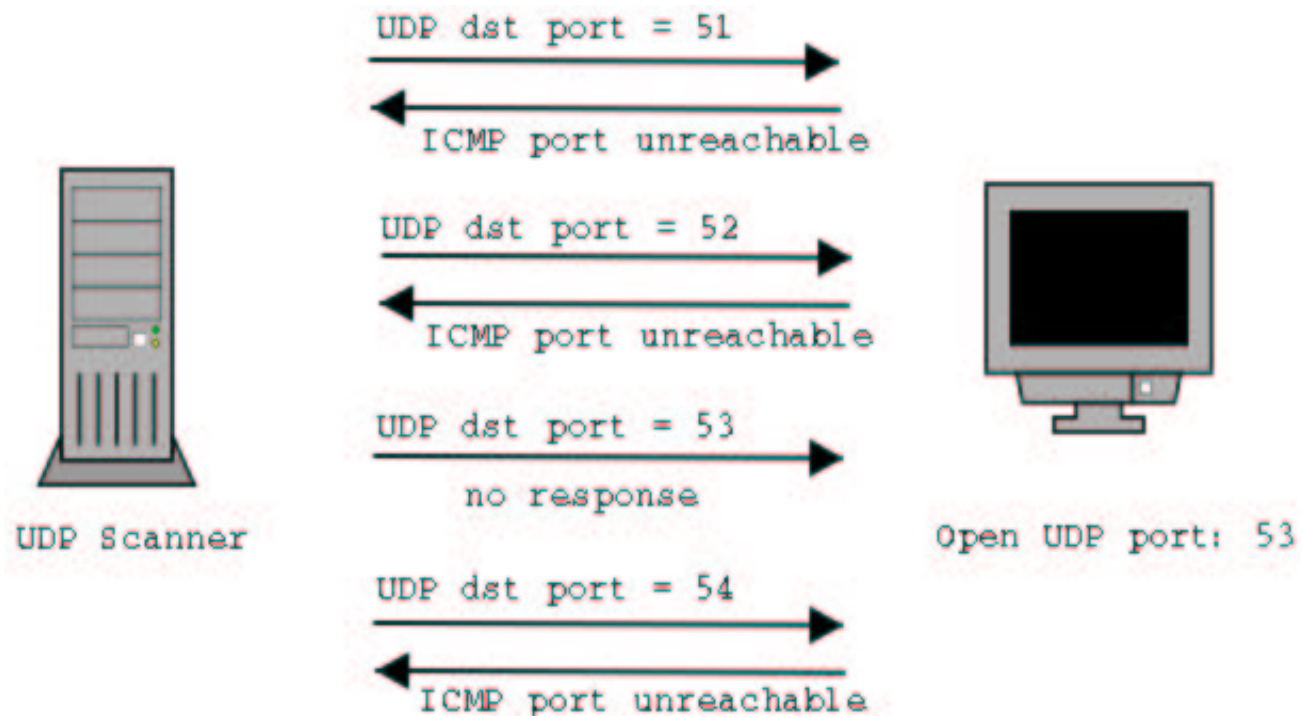
Why is this important again?

- Domain Name System (DNS)
- Trivial File Transfer Protocol (TFTP)
- Remote Authentication Dial In User Services (RADIUS)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)
- Network Time Protocol (NTP)
- Dynamic Host Configuration Protocol (DHCP)

UDP message format



UDP port probing



TCP and ICMP scanning

- TCP
 - 3-way handshake and reliability
 - Lots of header
 - Ever compare UDP and TCP RFCs?
 - See *nmap's* documentation
- ICMP
 - Request/reply messages
 - Lots of messages
 - Implementations differ widely
 - See Ofir Arkin's ICMP paper

The trouble with UDP scanning

From RFC 1122, Requirements for Internet Hosts, section 3.2.2.1:

A host SHOULD generate Destination Unreachable messages with code:

- 2 (Protocol Unreachable), when the designated transport protocol is not supported; or
- 3 (Port Unreachable), when the designated transport protocol (e.g., UDP) is unable to demultiplex the datagram but has no protocol mechanism to inform the sender.

Other failure scenarios

- Packet filtering
- Non-default host configurations
- Packet loss
- Errored packets
- ICMP rate limiting (see RFC 1812 section 4.3.2.8)

Minimizing false positives

- Verify ICMP replies
- Congestion avoidance
- Round trip time estimation
- See SATAN source code
- Implement application level scanning

UDP application scanning

- Solicit application layer replies
 - Most UDP apps will respond to something
- Few general purpose UDP application scanners
 - Most are for specific application vulnerabilities
- UDP application scanning has failure modes too
 - Which UDP port to scan?
 - How to format the message?
- So... I'm no Wietse, but what the heck I tried...

Application scanning examples

- Send a TFTP read request and check for error
- Send an empty RIP request with metric of infinity
- Send a version=[3|4] and mode=client NTP request
- App scanning for syslog would be useful, but alas...
- Other interesting applications?
 - e.g. games, streaming audio/video, trojans
- Most apps should be very easy to scan for
 - Just format the right request and await a reply

Is it Mai Tai time yet?

- UDP scanning is a relatively simple procedure
- However, be aware of how unreliable it is
- UDP application specific scanners would be better
- Application scanning may highlight vulnerabilities
- If not, PROTOS style projects certainly will