Frustrating OS Fingerprinting with Morph

Kathy Wang
Syn Ack Labs
knwang@synacklabs.net

Areas Covered in Talk

- OS Fingerprinting History
- What is Morph?
- Morph dependencies
- Morph architecture
- Implementation considerations
- Future directions
- Acknowledgments

What is OS Fingerprinting?

- Banner information
- Manual reconnaissance
- Active fingerprinting
- Passive fingerprinting
- Timing analysis fingerprinting

OS Fingerprinting History

- QueSO by Apostels
- Nmap by Fyodor
- p0f by Michael Zalewski
- Xprobe/Xprobe2 by Ofir Arkin and Fyodor Yarochkin
- RING by Franck Veysset, et al

Why Defeat OS Fingerprinting?

- Most attacks begin with some form of reconnaissance
- Target host OS information is very important
- OS scanners are designed to exploit expected OS behavior
- OS "honesty" leads to its own demise
- Not entirely vendors' faults

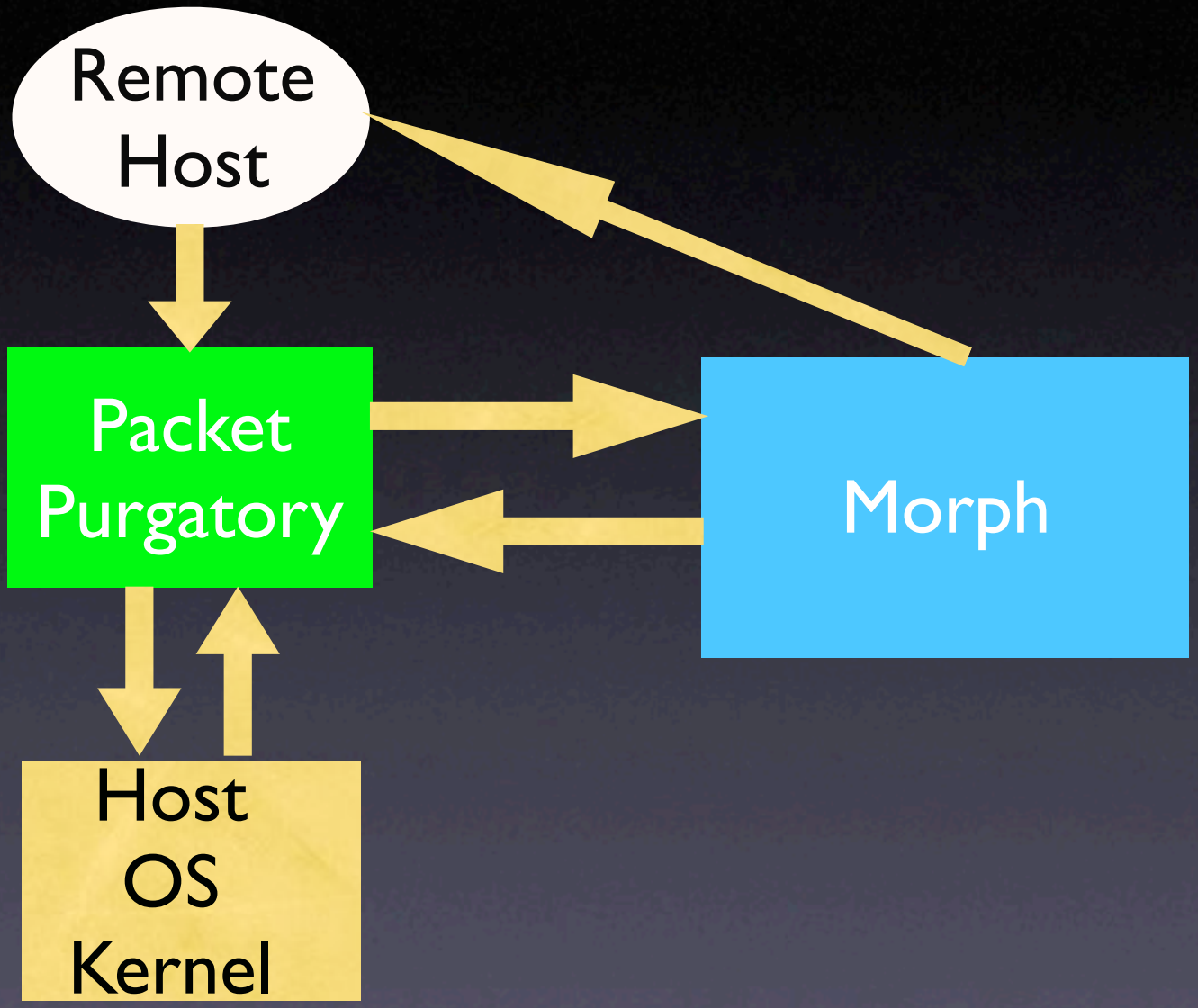
What is Morph?

- Morph is a process that allows user to select desired OS to emulate
  - Goal: Windows 2000 SP4, Linux 2.4.x.x, OpenBSD 3.3
- Will handle inbound and outbound packets and change TCP, UDP, ICMP and IP headers to reflect selected OS
- Morph is a tool that will currently compile on Linux, and is in development for OpenBSD, FreeBSD, NetBSD
- Not production quality yet
- BSD licensed
- Download at http://www.synacklabs.net/projects/morph

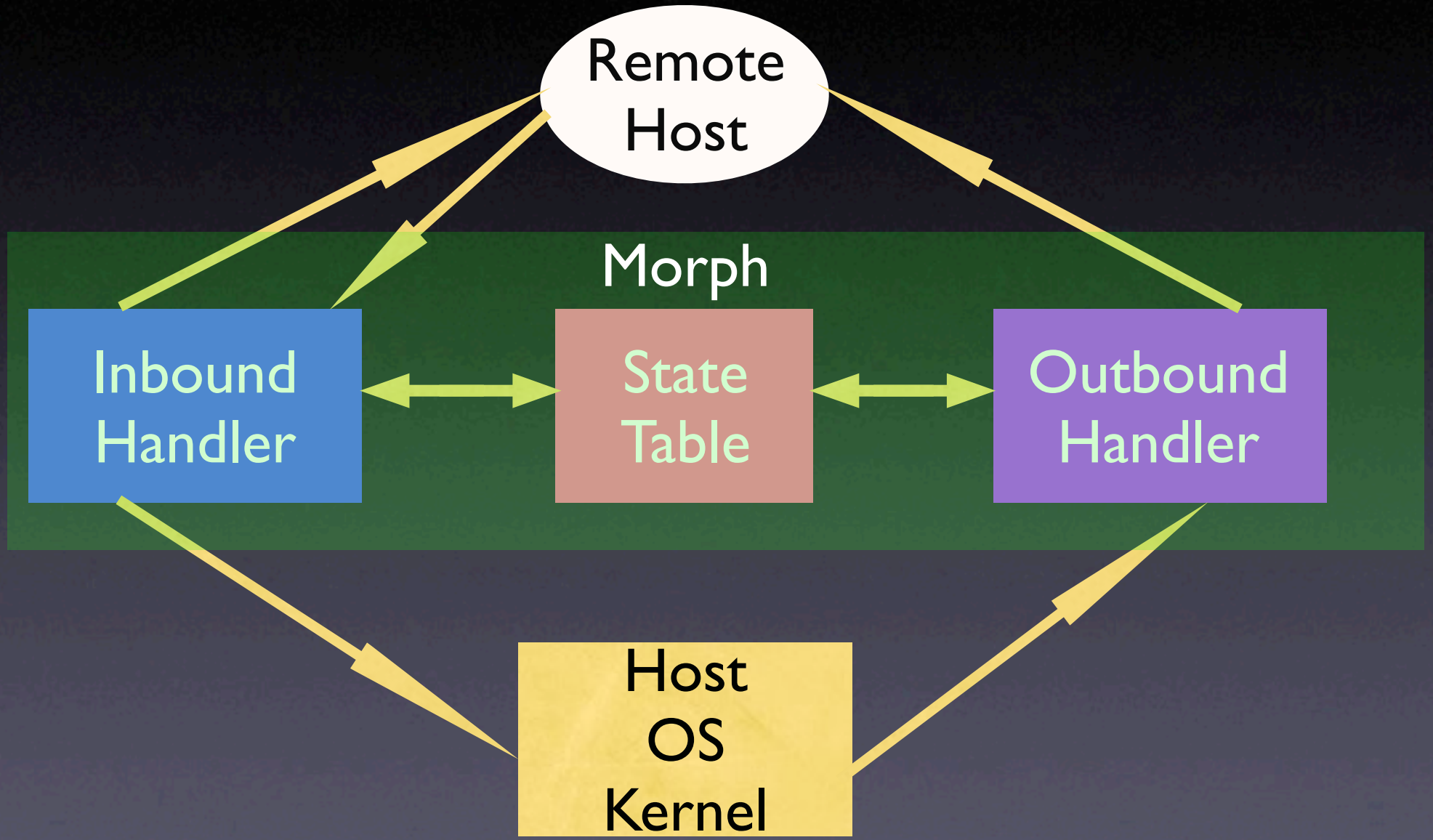
Morph Dependencies

- Morph is built on Packet Purgatory library
  - Wedge between OS kernel and network interface running in userland
- Packet Purgatory is built on libpcap and libdnet libraries
  - libpcap and libdnet provides interfaces to the kernel

High-Level Morph Architecture



Morph Internal Architecture



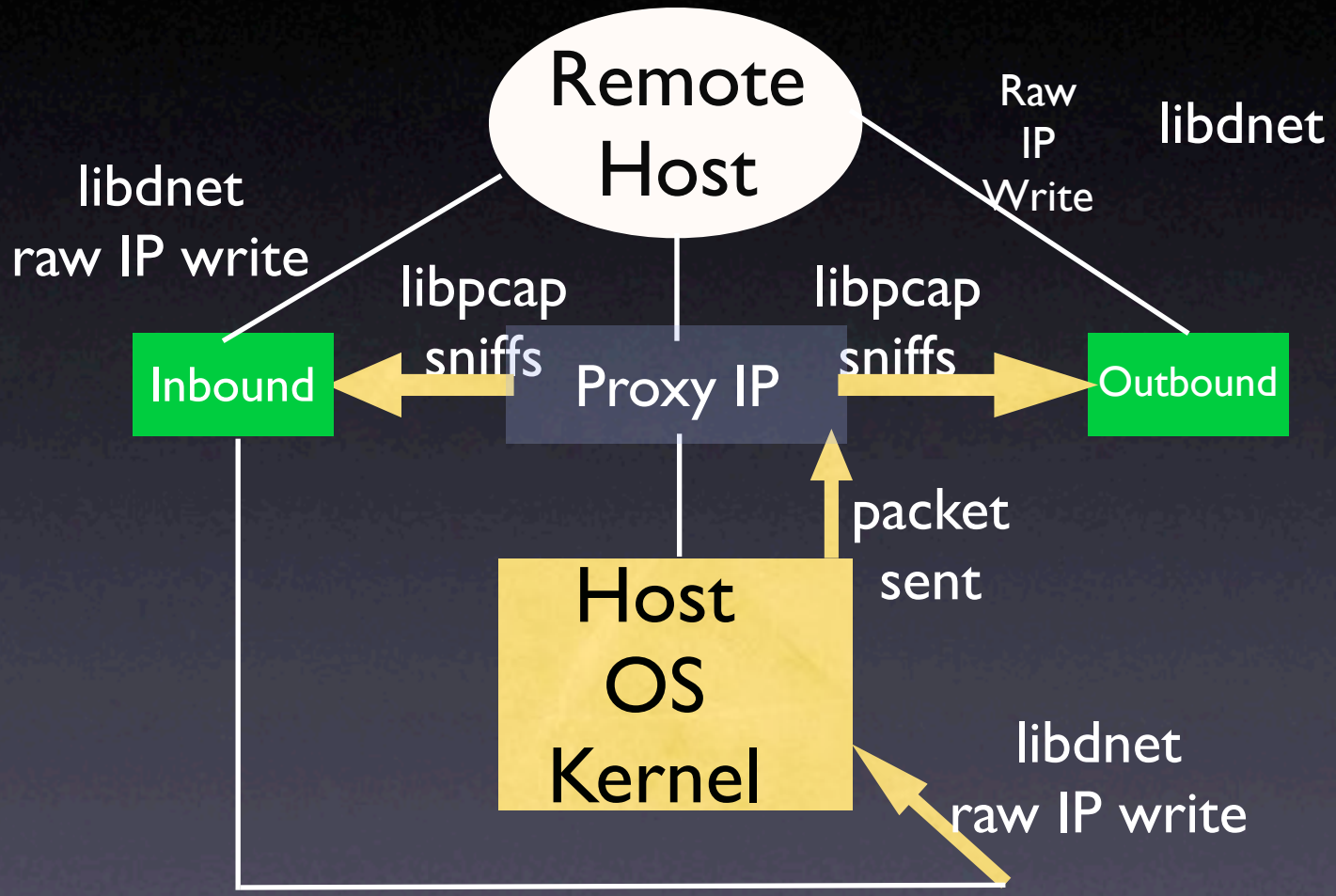
More About Packet Purgatory

- Route table maintains IP address to intercept messages to/from
- OS firewall prevents kernel from knowing about packets until done with tampering
- Not a kernel module
- BSD licensed
- http://www.synacklabs.net/projects/packetp

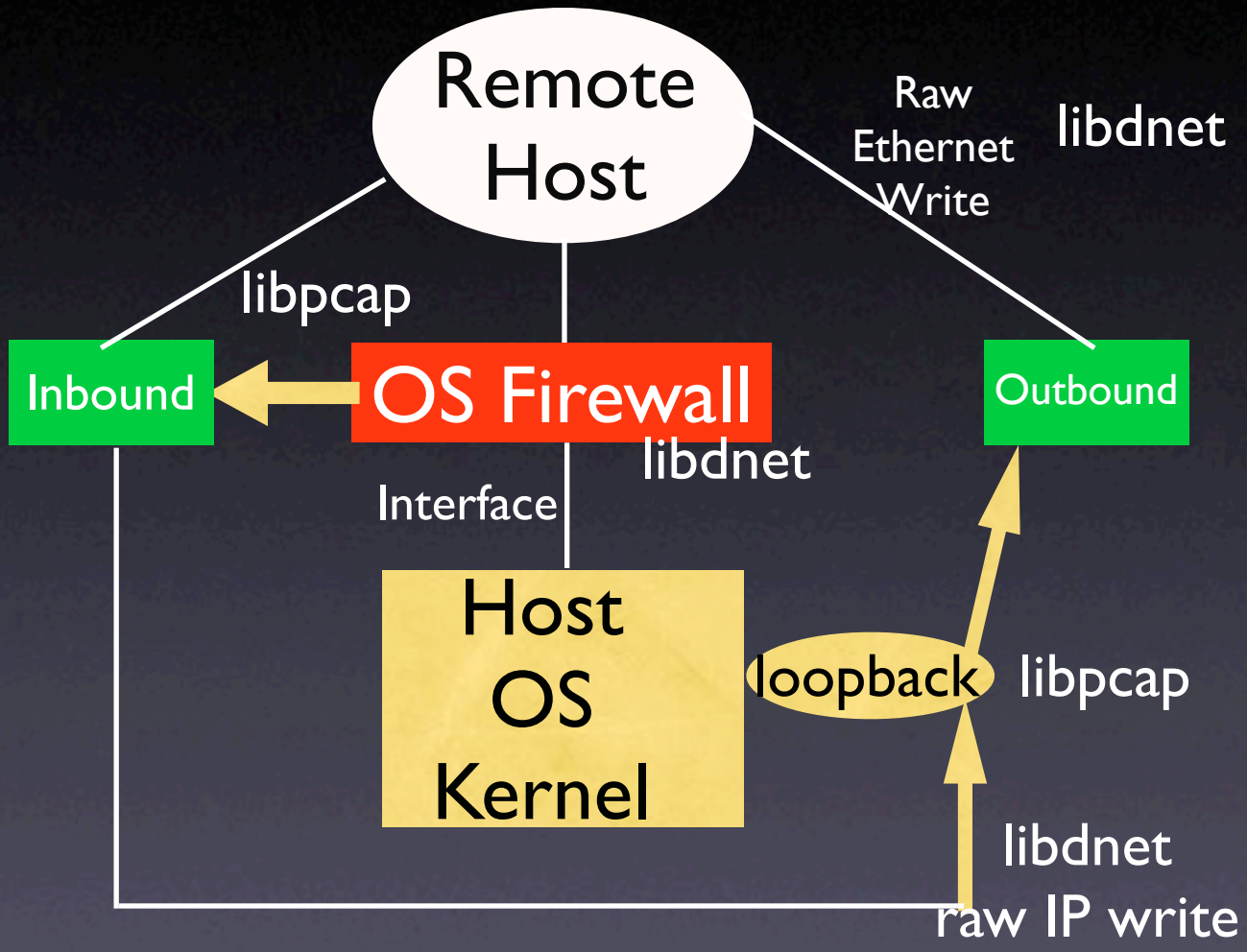
How Does Packet Purgatory Utilize libpcap and libdnet?

- Packet Purgatory has two modes
  - Proxy mode
  - Loopback-firewall mode

Proxy Mode


Remote Host
libdnet raw IP write
Raw IP Write
libdnet
libpcap sniffs
libpcap sniffs
Inbound
Proxy IP
Outbound
packet sent
Host OS Kernel
libdnet raw IP write

Loopback-Firewall Mode



OS scanners that Morph will fool

- QueSO
- Nmap
- Xprobe/Xprobe2
- p0f (in progress)
- RING/Snacktime (in progress)

Other Tools that Defeat OS Fingerprinting

- FPF
  - LKM for Linux
- IP Personality
  - Patch for Linux 2.4 kernel

- There are a couple of other tools
- None are highly portable
- Most will not emulate another OS

Current OS Fingerprinting Techniques

- Active fingerprinting
- Passive fingerprinting
- Timing analysis fingerprinting

- All of the above can be defeated with Morph

How does QueSO work?

- Utilizes active fingerprinting techniques
- Sends 7 different types of packets to open ports on target host
- All 7 packets sent modify TCP headers (e.g., different flags are set)
- OS fingerprint signatures are somewhat outdated (e.g., no Linux fingerprint beyond 2.1 kernel)

Morph Response to QueSO

	Morph Handling Status		
QueSO Packet Types	Inbound	State Table	Outbound
SYN	If port is open pass packet to OS, else write RST as a response	Add SYN connection	Rewrite packet to reflect emulated OS
SYN+ACK	Check state table to see if connection is a response	Will update table if packet is solicited	If packet is solicited, then write appropriate ACK reply
FIN	Pass packet to OS, or in cases of Windows-like behavior, reply	Don't care	Rewrite packet to reflect desired OS
FIN+ACK	Respond on behalf of emulated OS	Don't care	Don't care
SYN+FIN	Respond on behalf of emulated OS	Don't care	Don't care
PSH	Pass packet to OS	Don't care	Rewrite packet to reflect desired OS
SYN+XXX+YYY	Depending on emulated OS, respond on behalf of emulated OS	Possibly add SYN connection	May rewrite packet to reflect emulated OS

How does Xprobe2 work?

- Utilizes active fingerprinting techniques
- Xprobe2 sends 4 different types of ICMP packets to target host
- Information request packet is basically obsolete (W. Richard Stevens, TCP/IP Illustrated, Vol. 1)
- UDP packet is sent for ICMP unreachables
- Final packet is vanilla SYN

Morph Response to Xprobe2 0.2

	Morph Handling Status		
Xprobe2 Packet Types	Inbound	State Table	Outbound
ICMP ECHO	Respond on behalf of emulated OS	Don't care	Don't care
ICMP Timestamp	Respond on behalf of emulated OS	Don't care	Don't care
ICMP Address Mask Request	Respond on behalf of emulated OS	Don't care	Don't care
ICMP Information Request	Respond on behalf of emulated OS	Don't care	Don't care
UDP -> ICMP Unreachable (Includes UDP Port Unreachable Error Message)	If port probed is open, pass to OS. Otherwise, respond on behalf of emulated OS	Don't care	Rewrite appropriate reply according to emulated OS
TCP SYN (Includes TCP Header Information)	If port is open pass packet to OS, else write RST as a response	Add SYN connection	Rewrite packet to reflect emulated OS

How does Nmap work?

- Nmap sends 9 different types of packets to target host
- Needs both open and closed ports for accuracy
- Nmap is challenging to defeat
  - Nmap uses many test cases
  - Sends non-standard, non-documented packet types to pinpoint OS of target

Morph Response to Nmap 3.50

		Morph Handling Status		
Nmap Packet Types		Inbound	State Table	Outbound
Open Port	TCP Sequence Test	Pass packet to OS	Add SYN connection	Send response packet to reflect emulated OS
	SYN with Options	Pass packet to OS	Add SYN connection	Send response packet to reflect emulated OS
	NULL with Options	Respond on behalf of emulated OS	Don't care	Don't care
	SYN-FIN-URG-PSH with Options	If OS accepts it, pass to OS. Otherwise, respond on behalf of emulated OS	Add connection	If applicable, send response to reflect emulated OS
	ACK with Options	If connection exists, pass packet to OS. Otherwise, respond on behalf of emulated OS	If part of existing connection, add ACK connection	Send response packet to reflect emulated OS if part of existing connection
Closed Port	SYN with Options	Respond on behalf of emulated OS	Don't care	Don't care
	ACK with Options	Respond on behalf of emulated OS	Don't care	Don't care
	PSH-FIN-URG with Options	Respond on behalf of emulated OS	Don't care	Don't care
	UDP Packet	Respond on behalf of emulated OS	Don't care	Don't care

Morph State Table

- Remote host sends packet
- Morph generates a “random” sequence number based on emulated OS
- Morph state table maintains session sequence number offset information
- Sequence number gets modified on the way to remote OS

Fooling other OS scanners

- p0f (passive OS fingerprinting)
- RING (packet timing analysis)
- Snacktime (packet timing and passive analysis)

New OS Fingerprinting Techniques

- CanSecWest talk on new OS fingerprinting techniques
- Instead of sending single packet to solicit response, sends multiple packets
  - Uses layer 7 info
  - Expands timing analysis
  - Measures window behavior under congested conditions

How can you avoid being fingerprinted?

- New RFC needed to address currently unspecified behavior
- Place hardened critical servers behind intermediate proxying devices

Challenges to Defeating OS Fingerprinting

- Advertising different window size than what underlying OS support
- Having to maintain state of connections to distinguish between normal vs abnormal connections
- Not necessarily having access to standard implementations, and having to glean information through fingerprints
- Even if responses to OS scanners are accurate, application scanning can reveal true OS (implement PolyMorph)
- Some automated attacks do not care what OS it's attacking (NIMDA)

Future Directions for Morph

- Support more operating system emulation (Solaris, HP-UX, etc)
- Support Morph installs on more operating systems (Windows 2000/XP)
- Fool other OS scanners (p0f, RING, etc)
- Fool application scanners (PolyMorph)
- Add GUI support for Morph

Acknowledgments

- Todd MacDermid
- Bill Neugent
- Don Bailey
- Dan Aiello
- Dave Wilburn
- Bob Fleck
- Dave Dandar

Questions?