Hacking Exposed 7
Network Security Secrets & Solutions

Chapter 2 Scanning

Scanning

- Determining if the system is alive
- Determining which services are running or listening
- Detecting the operating system
- Processing and storing scan data

Determining If the System is Alive

- Network ping sweeps
  - ARP host discovery: on the same subnet
    - Arp-scan: run as root by sudo to list IP-MAC
    - Nmap (Network Mapper): host and service discovery with various options (host only: -PR –sn)
    - Cain (Windows-only): beyond host and service discovery

Address Resolution Protocol (ARP)

- root@kali:~# arp-scan --interface=wlan0 –localnet
- Interface: wlan0, datalink type: EN10MB (Ethernet)
- Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/)
- 10.0.1.3 0 b:1a:a0:c2:94:c0 Dell Inc
- 10.0.1.57 0b:0c:29:34:f9:6a VMware, Inc.
- 10.0.1.253 0b:19:55:9d:60:c1 CISCO SYSTEMS, INC.
- 29 packets received by filter, 0 packets dropped by kernel Ending arp-scan 1.9: 256 hosts scanned in 2.259 seconds (113.32 hosts/sec). 29 responded

Nmap –sn –PR –send-IP <IP-range>

- NMAP not only sends an ICMP ECHO REQUEST packet it also performs an ARP ping, some TCP pinging.
- **Understanding what tools do, is really important**. If the target network is being monitored by an IDS, you may inadvertently trigger an alert because of all of the extra traffic being generated

Determining If the System is Alive

- Network ping sweeps
  - ARP host discovery: on the same subnet
    - Arp-scan: run as root by sudo to list IP-MAC
    - Nmap (Network Mapper): host and service discovery with various options (host only: -PR –sn)
    - Cain (Windows-only): beyond host and service discovery
  - ICMP host discovery: remote host/router
    - ICMP ECHO REQUEST, ICMP ECHO REPLY, ICMP TIMESTAMP, ICMP ADDRESS MASK, etc.
    - Ping: OS utilities for ECHO REQUEST/REPLY
    - Nmap: ICMP ping/address mask/timestamp, ARP ping, TCP ping
    - Hping3 and nping: any combinations of flags on any combinations of packet types, spoofing MAC/IP
    - Superscan: multiple ICMP in parallel
  - TCP/UDP host discovery: when internal and/or external ICMP is not permitted
    - Servers: TCP/UDP service ports
    - Desktops: local firewall to ban inbound connections, but accessible through remote desktop, file sharing, and disabled local firewall
    - Nmap/Superscan/Nping: all ports (slow and noisy) or specific ports

```
user@hax:~$ sudo nmap -sn -PE --send-ip 192.168.1.1

Starting Nmap 5.51 (http://nmap.org) at 2011-09-24 10:06 PDT
Nmap scan report for 192.168.1.1
Host is up (0.060s latency).
MAC Address: 5F:8D:09:F4:07:43 (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

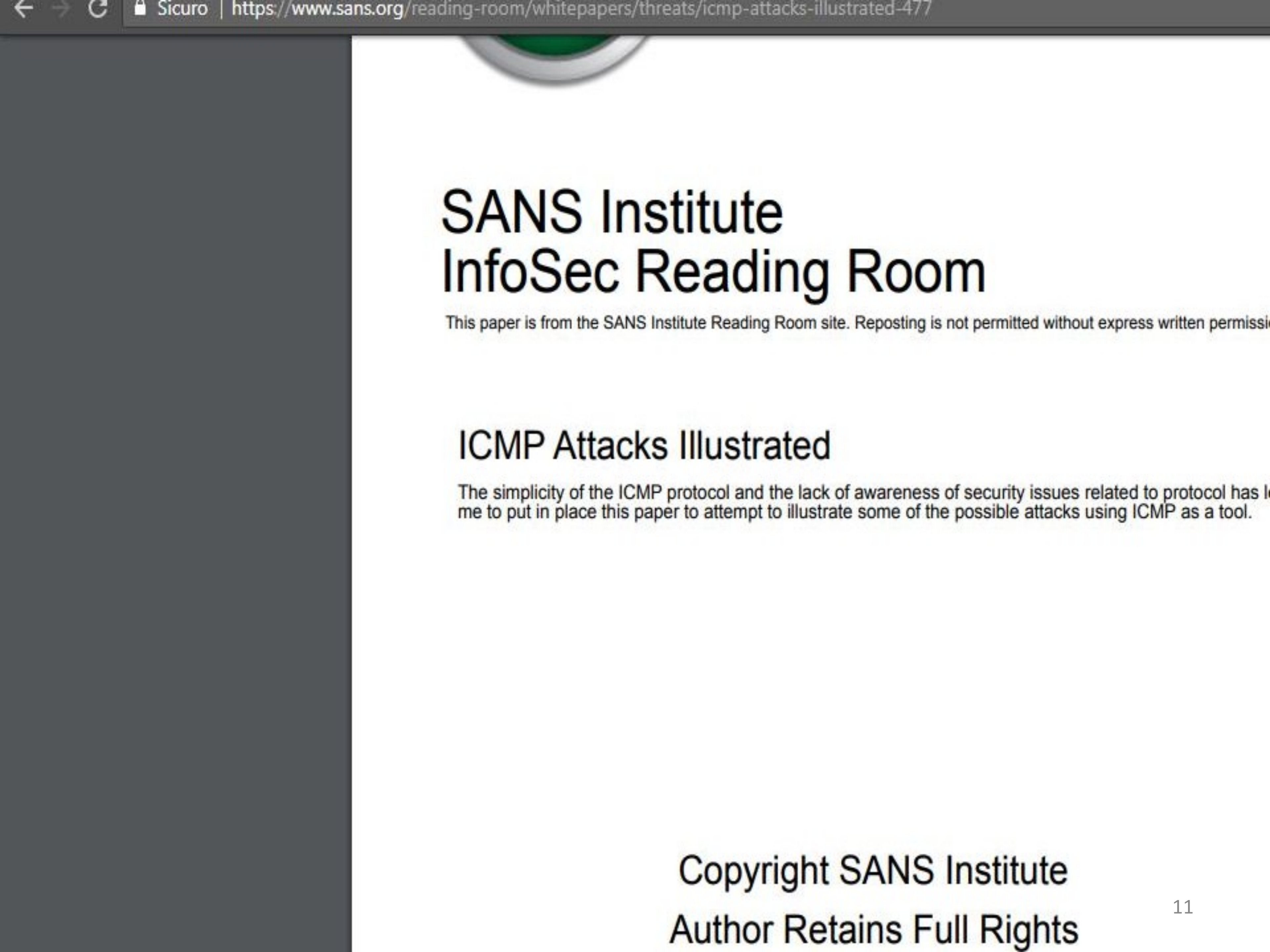
```
user@hax: ~$ ping -c 2 192.168.1.1

PING 192.168.1.1 (192.168.1.1) 56(84) bytes Of data.
64 bytes from 192.168.1.1: icmp_req=1 tt1=64 time=0.149 ms
64 bytes from 192.168.1.1: icmp_reg=2 tt1=64 time=0.091 ms
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.091/0.120/0.149/0.029
```

Determining If the System is Alive

- Network ping sweeps
  - ARP host discovery: on the same subnet
    - Arp-scan: run as root by sudo to list IP-MAC
    - Nmap (Network Mapper): host and service discovery with various options (host only: -PR –sn)
    - Cain (Windows-only): beyond host and service discovery
  - ICMP host discovery: remote host/router
    - ICMP ECHO REQUEST, ICMP ECHO REPLY, ICMP TIMESTAMP, ICMP ADDRESS MASK, etc.
    - Ping: OS utilities for ECHO REQUEST/REPLY
    - Nmap: ICMP ping/address mask/timestamp, ARP ping, TCP ping
    - Hping3 and nping: any combinations of flags on any combinations of packet types, spoofing MAC/IP
    - Superscan: multiple ICMP in parallel
  - TCP/UDP host discovery: when internal and/or external ICMP is not permitted
    - Servers: TCP/UDP service ports
    - Desktops: local firewall to ban inbound connections, but accessible through remote desktop, file sharing, and disabled local firewall
    - Nmap/Superscan/Nping: all ports (slow and noisy) or specific ports

Ping Sweeps Countermeasures

- Detection
  - IDS: snort
  - Commercial firewall: network or desktop
    - Detect ICMP, TCP, UDP ping sweeps
      - A pattern of ICMP/TCP/UDP packets from a particular system or network
  - Host based tools: Scanlogd, courtney, ippl, protolog
  - **Not just tools, eyeballs count.**
- Prevention
  - ACL in firewall: limit ICMP traffic into your networks or systems
  - Allow only ECHO_REPLY, HOST UNREACHABLE, TIME EXCEEDED into specific hosts in DMZ; allow only ISP's specific IP addresses
    - Loki2: hackers use it to backdoor the OS and tunnel data in ICMP ECHO
  - Pingd: move ICMP from kernel to user space

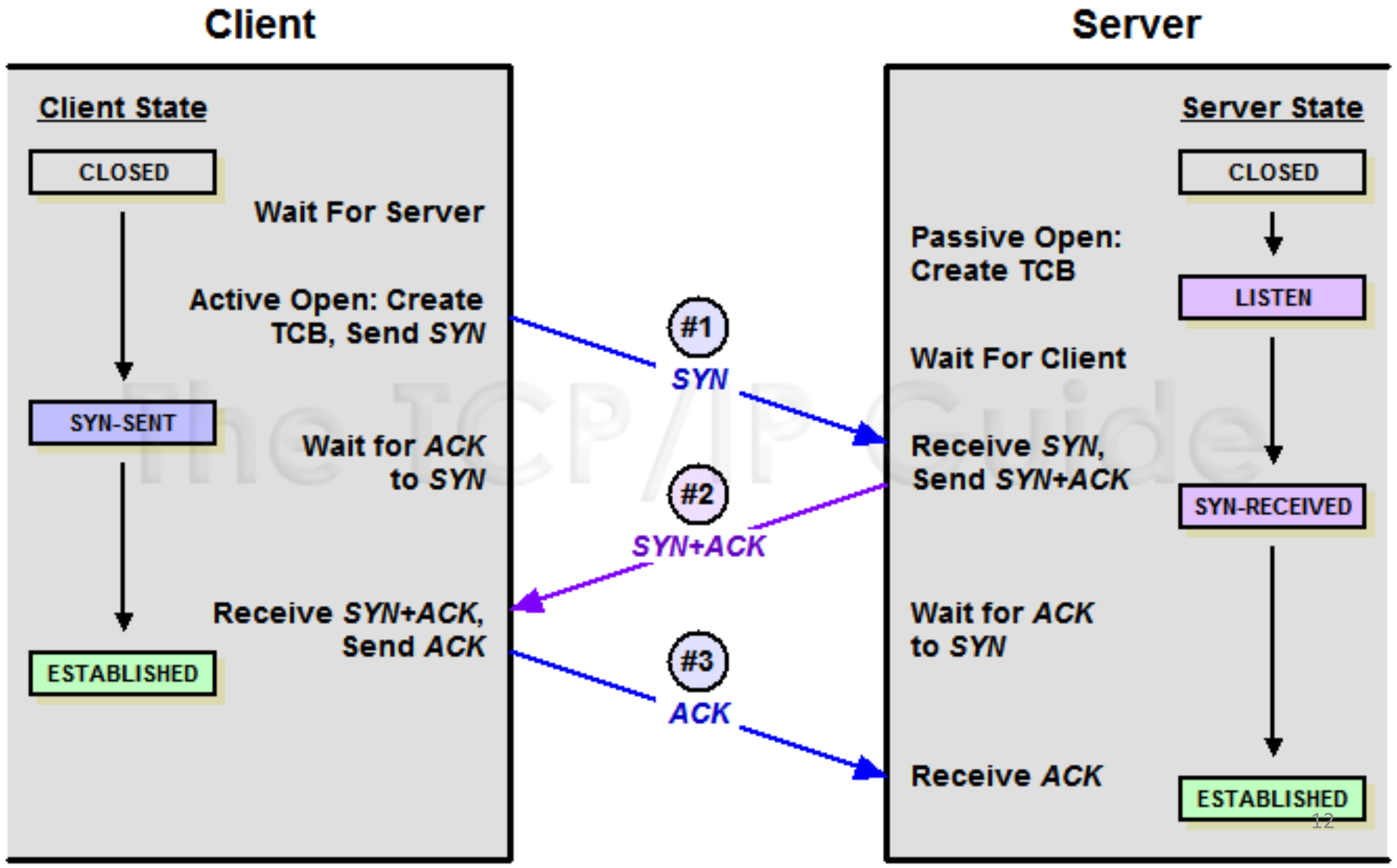

Sicuro
https://www.sans.org/reading-room/whitepapers/threats/icmp-attacks-illustrated-477

SANS Institute
InfoSec Reading Room

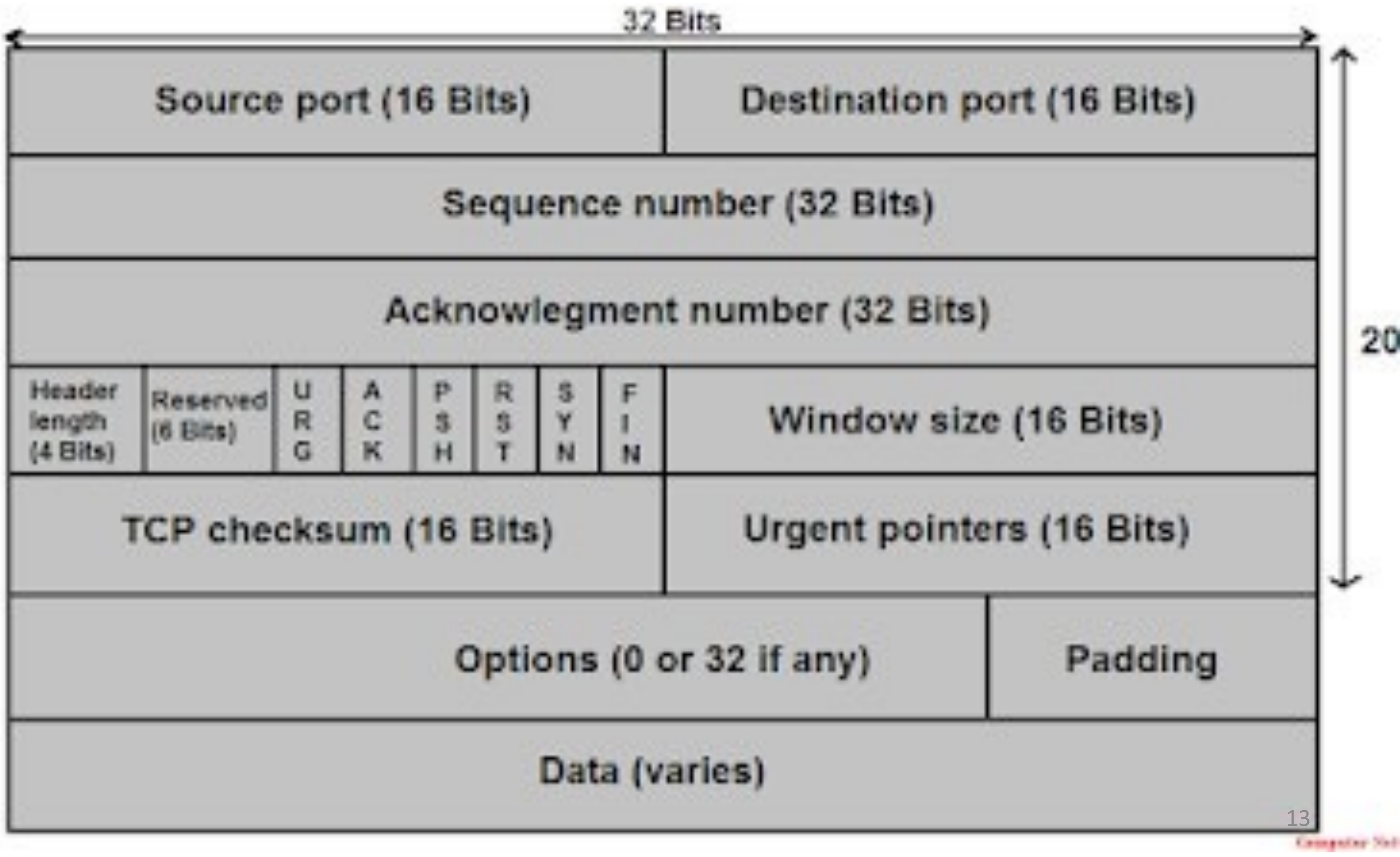
ICMP Attacks Illustrated

The simplicity of the ICMP protocol and the lack of awareness of security issues related to protocol has l
me to put in place this paper to attempt to illustrate some of the possible attacks using ICMP as a tool.

3-way handshake



TCP Header



32 Bits
Source port (16 Bits)
Destination port (16 Bits)
Sequence number (32 Bits)
Acknowlegment number (32 Bits)
Header length (4 Bits)
Reserved (6 Bits)
URG
ACK
PSH
RST
SYN
FIN
Window size (16 Bits)
TCP checksum (16 Bits)
Urgent pointers (16 Bits)
Options (0 or 32 if any)
Padding
Data (varies)
20

TCP FLAG

- URG (1 bit): indicates that the Urgent pointer field is significant
- ACK (1 bit): indicates that the Acknowledgment field is significant. All packets after the initial SYN packet sent by the client should have this flag set.
- PSH (1 bit): Push function. Asks to push the buffered data to the receiving application.
- RST (1 bit): Reset the connection
- SYN (1 bit): Synchronize sequence numbers. Only the first packet sent from each end should have this flag set. Some other flags and fields change meaning based on this flag, and some are only valid for when it is set, and others when it is clear.
- FIN (1 bit): Last packet from sender.

Determining Which Services Are Running or Listening

- Port scanning
  - Identifying TCP/UDP services running on the target
  - Identifying type of OS of the target
  - Identifying applications or versions of a service
  - Scan types (anomalous TCP packets)
    - TCP connect scan (3-way handshake)
    - TCP SYN scan (half-open scan, SYN then SYN/ACK or RST/ACK) (not traceable)
    - TCP FIN scan (RST if closed port)
    - TCP Xmas Tree scan (FIN/URG/PUSH)
    - TCP null scan, TCP ACK scan
    - TCP Windows scan
    - TCP RPC scan
    - UDP scan (ICMP port unreachable msg, if closed port)

Detecting The Operating System

Active Operating System Detection

- Useful info for vulnerability mapping
  - Banner grabbing: some applications tell it all
    - FTP, telnet, SMTP, HTTP, POP, and other
  - Scanning available ports: some services are OS specific!
  - Stack fingerprinting: TCP/IP stack implementation

Detecting The Operating System

Active Operating System Detection

- Making guess from available ports
  - Windows: ports 135, 139, 445 (139 only for Windows 95/98); 3389 for RDP (Remote Desktop Protocol)
  - UNIX: TCP 22 (SSH), TCP 111 (RPC portmapper=port 135), TCP 512-514 (Berkeley Remote services, rlogin), TCP 2049 (NFS, Network File System), high number ports 3277x

    (**RPC, Remote Procedure Call in Solaris**)

Port 135, 139, 445

- Port 135 Microsoft EPMAP, end-point mapper. Microsoft relies upon DCE RPC to remotely manage services. Some services that use port 135 of end-point mapping are: DHCP server, DNS server, WINS server
- Port 139 NetBIOS
- Port 445 MS Server Message Block (SMB), SAMBA-compatible

Detecting The Operating System
Active Operating System Detection

- Active stack fingerprinting (Phrack Magazine)
  - Vendors interpret RFCs differently when writing TCP/IP stack
  - Nmap –O: signature listing at Nmap-os-fingerprints
    - **FIN probe (**Windows 7/200x/Vista respond with FIN/ACK)
    - **Bogus flag probe** (Linux responds with the same undefined flag)
    - **Initial Sequence Number sampling** (find pattern in the ISSN)
    - **"Don't fragment bit" monitoring** (some OS have it set)
    - **TCP initial window size** (tracked on returned packets)
    - **ACK value** (same number sent +0 or +1)
    - **ICMP message quenching** (RFC 1812 limit rate of error messages)
    - **ICMP message quoting** (# of quoted info in errors)
    - **ICMP message echoing integrity** (IP headers changed in ICMP errors)
    - **Type of service (TOS)** (most implementation use 0, but can vary)
    - **fragmentation handling**
    - **TCP options**

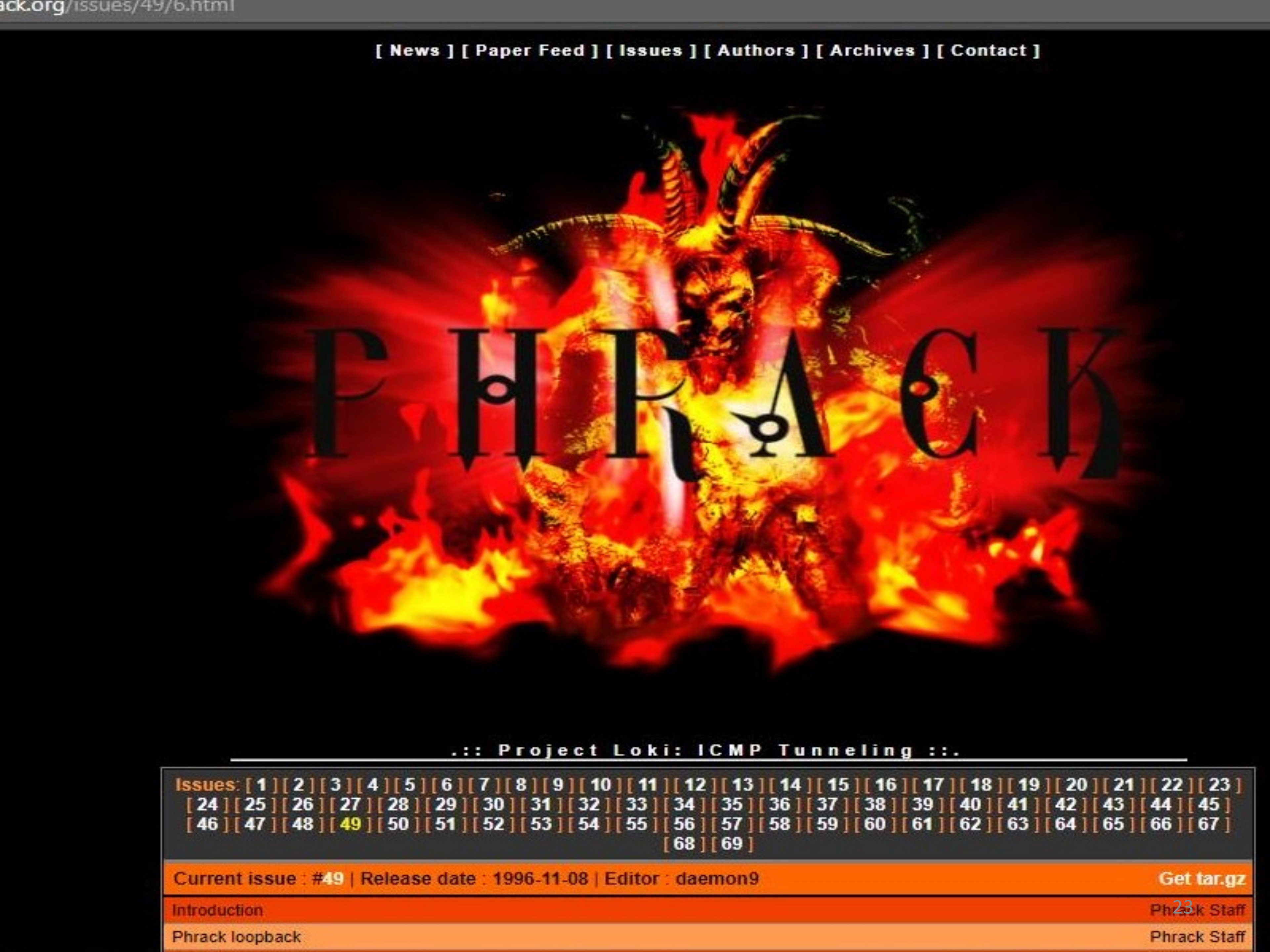
```
user@hax:~$ sudo nmap -O 192.168.1.17

Starting Nmap 5.51 ( http://nmap.org ) at 2011-09-26 11:35 PDT
Nmap scan report for 192.168.1.17
Host is up (0.0015s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-term-serv
4445/tcp  open  upnotifyp
14000/tcp open  scotty-ft
Device type: general purpose
Running: Microsoft Windows XP
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://
nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 3.64 seconds
```

Detecting The Operating System
Active Operating System Detection

- Countermeasures
  - **Detection**
    - They can detect scan with an option set (e.g. SYN)
    - Use Snort, Scanlogd, ecc.
  - **Prevention**
    - Change unique stack characteristic (not recommended)
    - secure proxy or firewall, Active Defence

Detecting The Operating System
Active Operating System Detection

- Useful info for vulnerability mapping
  - Banner grabbing: some applications tell it all
  - Scanning available ports: some services are OS specific!
  - Stack fingerprinting: TCP/IP stack implementation
- Making guess from available ports
  - Windows: ports 135, 139, 445 (139 only for Windows 95/98); 3389 for RDP (Remote Desktop Protocol)
  - UNIX: TCP 22 (SSH), TCP 111 (RPC portmapper=port 135), TCP 512-514 (Berkeley Remote services, rlogin), TCP 2049 (NFS, Network File System), 3277x (**RPC, Remote Procedure Call in Solaris**)
- Active stack fingerprinting (Phrack Magazine)
  - Vendors interpret RFCs differently when writing TCP/IP stack
  - Nmap –O: signature listing at Nmap-os-fingerprints
    - FIN probe (Windows 7/200x/Vista respond with FIN/ACK), Bogus flag probe, Initial Sequence Number sampling, “Don’t fragment bit” monitoring, TCP initial window size, ACK value (+0 or +1), ICMP message quenching, ICMP message quoting, ICMP message echoing integrity, TOS, fragmentation handling, TCP options
- Countermeasures
  - Detection: same detection tools: Snort, Scanlogd, ecc.
  - Prevention: secure proxy or firewall, Active Defence



ack.org/issues/49/6.html

[News] [Paper Feed] [Issues] [Authors] [Archives] [Contact]

.:: Project Loki: ICMP Tunneling ::.

Issues: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69]

Current issue : #49 | Release date : 1996-11-08 | Editor : daemon9 — Get tar.gz

Introduction	Phrack Staff
Phrack loopback	Phrack Staff

TCP/IP Stack Fingerprint

The TCP/IP fields that may vary include the following:

- Initial packet size (16 bits)
- Initial TTL (8 bits)
- Window size (16 bits)
- Max segment size (16 bits)
- Window scaling value (8 bits)
- "don't fragment" flag (1 bit)
- "sackOK" flag (1 bit)
- "nop" flag (1 bit)

These values may be combined to form a 67-bit signature, or fingerprint, for the target machine. Just automatically checking the Initial TTL and window size fields is often enough in order to successfully identify an operating system.

Detecting The Operating System

Passive Operating System Detection

- To be stealthy to IDS: passive
- Passive stack fingerprinting
  - At a central location or a port with packet capture (by port mirroring)
  - Siphon: a passive port-mapping, OS identification, and network topology tool
    - Passive signatures in osprints.conf
      - TCP/IP session: TTL, window size, DF (Don’t Fragment), etc.
  - Tend to fail if: (1) applications build their own packets, (2) not able to capture packets, (3) a remote host changes the connection attributes (active detection also fails on this)
- Countermeasures
  - Same as OS detection countermeasures

- First, we telnet from the system shadow (192.168.1.10) to quake (192.168.1.11)

```
[shadow]# telnet 192.168.1.11
```

- Then, sniff packet using **snort**

```
06/04-11:23:48.297976 192.168.1.11:23 →> 192.1681.10:2295
TCP TTL:255 TOS:0×0 ID:58934 DF
**S***A* Seq: 0xD3B709A4 Ack: 0xBE09B2B7 Win: 0x2798
TCP Options => NOP NOP TS: 9688775 9682347 NOP WS: 0 MSS: 1460
```

- Let us look at the siphon signature fingerprinting database

```
[shadow] # grep -i solaris osprints.conf
# Window:TTL:DE: Operating System DE = 1 for ON, 0 for OFF.
2328:255:1:Solaris 2.6 - 2.7
2238:255:1:Solaris 2.6 - 2.7
2400:255:1:Solaris 2.6 - 2.7
2798:255:1:Solaris 2.6 - 2.7
FE88:255:1:Solaris 2.6 - 2.7
87C0:255:1:Solaris 2.6 - 2.7
FAFO:255:0: Solaris 2.6 - 2.7
FEFF:255:1:Solaris 2.6 - 2.7
```

- If we use **siphon**, we can see the OS matching as Solaris

```
[crush]# siphon -v -i x10 -o fingerprint.out
Running on: 'crush' running FreeBSD 4.0-RELEASE on a(n) i386
Using Device: x10
Host                Port   TTL    DF     Operating System
192 168 1 11        23     255    ON     Solaris 2.6 - 2.7
```

Processing and Storing Scan Data

- Efficiency in managing scan data → speed to compromise a large number of systems
- Metasploit
  - A vast platform of tools, payload, and exploits
  - PostgreSQL for database
  - db_connect: tells metasploit how to connect to database and which database to use

```
msf > db connect postgres:<password>@localhost: <port>/msf3
```

  - db_nmap (root required): run Nmap scans
    - Metasploit could scan but slower than Nmap
  - db_import: import Nmap results into database, commands:
    - hosts: show hosts and their OS
    - services: show all available ports and services
    - Filtering (-s) to see, e.g., all hosts with SSH or running Windows 2008

```
msI > **db_nmap 192.168.1.0/24**
[*] Nmap: Starting Nmap 5.51SVN (http://nmap.org) at 2011-09-26 10:47 PDT

[*] Nmap: 22/tcp open ssh
[*] Nmap: Nmap done: 256 IP addresses (21 hosts up) scanned in 19.00 seconds
msf
[*] Nmap: Nmap scan report for 192.168.1.12
[*] Nmap: Host is up (0.0028s latency).
[*] Nmap: Not shown: 997 filtered ports
[*] Nmap: PORT            STATE    SERVICE
[*] Nmap: 80/tcp          open     http
[*] Nmap: 443/tcp         open     https
[*] Nmap: 2869/tcp        open     icslap

< Output shortened for brevity >

[*] Nmap: Nmap scan report for 192.168.1.13
[*] Nmap: Host is up (0.063s latency).
```

```
msi > sudo nmap -O 192.168.1.0/24 -oX subnet 192.168.1.0-OS
[*] exec: sudo nmap -O 192.168.1.0/24 -oX subnet 192.168.1.0-
OS
[sudol password for user:
Starting Nmap 5.51 (http://nmap.org) at 2011-09-26 11:00 PDT
Nmap scan report for 192.168.1.12
Host is up (0.0033s latency).
Not shown: 997 filtered ports
PORT   STATE  SERVICE
80/tcp open   http
< Output shortened for brevity >
OS details: Linux 2.6.19 - 2.6.36
Network Distance: 0 hops
msf >
```

```
ms > db import subnet 192.168.1.0-OS
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.4.3.1'
[*] Importing host 192.168.1.12
‹ Output shortened for brevity >
[*] Importing host 192.168.1.25
[*] Successfully imported
/home/elec/subnet_192. 168.1.0-OS msf>
```

Processing and Storing Scan Data

- Efficiency in managing scan data → speed to compromise a large number of systems
- Metasploit
  - A vast platform of tools, payload, and exploits
  - PostgreSQL for database
  - db_connect: tells metasploit how to connect to database and which database to use

```
msf > db connect postgres:<password>@localhost: <port>/msf3
```

  - db_nmap (root required): run Nmap scans
    - Metasploit could scan but slower than Nmap
  - db_import: import Nmap results into database, commands:
    - hosts: show hosts and their OS
    - services: show all available ports and services
    - Filtering (-s) to see, e.g., all hosts with SSH or running Windows 2008

```
msf > hosts -c address, os_name

Hosts
=======
Address        os_name
-------        -------
192.168.1.12  Microsoft Windows
192.168.1.15  Linux
192.168.1.16  Microsoft Windows
192.168.1.17  Microsoft Windows
192.168.1.18  Microsoft Windows
192.168.1.19  Apple iOS
192.168.1.22  Microsoft Windows
192.168.1.24  Microsoft Windows
192.168.1.25  Linux

msf > services -s ssh

Services
========
Host                port   proto  name   state  info
----                ----   ----   ----   ----   ----
10.112.18.25 22     22     tcp    ssh    open
```