Password Cracking and Sniffing

- Agenda
  - Storing Passwords on the system
  - Password Cracking on Windows and Linux
  - Defenses against Password cracking
  - Sniffing
  - Defenses against Sniffing

Cracking Passwords

- Passwords that can be guessed easily are a problem
- Lots of tools available to figure out passwords
- L0phtcrack windows password cracker
- “John the Ripper” Unix password cracker
- Default passwords remaining on a system are a typical vulnerability

Password storage

- Password files have passwords stored in a hashed or encrypted form
- Hash algorithm example is message digest 4 (MD4)
- Encrypted algorithm example is Data Encryption Standard (DES)
- When you use your password, it is hashed or encrypted and then compared to the stored value
- Crackers use a downloaded local copy of password file on their own machine

Storing Passwords

- Systems have a file with all hashed/encrypted passwords
  - Windows – SAM (Security Accounts Manager) database
  - UNIX - /etc/passwd or /etc/shadow
- Access to these files can make it easy for a hacker to break in

Windows Passwords

- Security Accounts Manager (SAM) has two versions for each password
- LanMan (LM) password version for backward compatibility with windows workgroups
- NT Hash – cryptographic hash for windows NT/2000 (Uses MD4)
- SAM file is in \WINNT\system32\config\ directory which is a binary file that is hard to read
- Back up copy stored in \WINNT\repair

Using Passwords

- System has a hashed/encrypted version of the password stored in a file
- On login attempt–
  - system hashes/encrypts the password typed in by using for example crypt() function in linux
  - Compares hashed/encrypted value to stored hashed/encrypted value
  - Idea behind password cracking is to get a copy of the hashed/encrypted passwords and then make guesses, hash/encrypt the guess and compare

Password Guessing

- Based on Dictionary
- Brute Force – Guess every possible combination of characters
- Hybrid – Use dictionary but add characters to dictionary entries

Password Cracking

- Dictionary Attack
  - Hackers steal a copy of the stored password file
  - Guess a password (may use a dictionary)
  - Find hash/encrypted value of the guess
  - Compare hash to entries from stored file
  - Continue this till success or out of options for password guesses.

Password retrieval on Windows

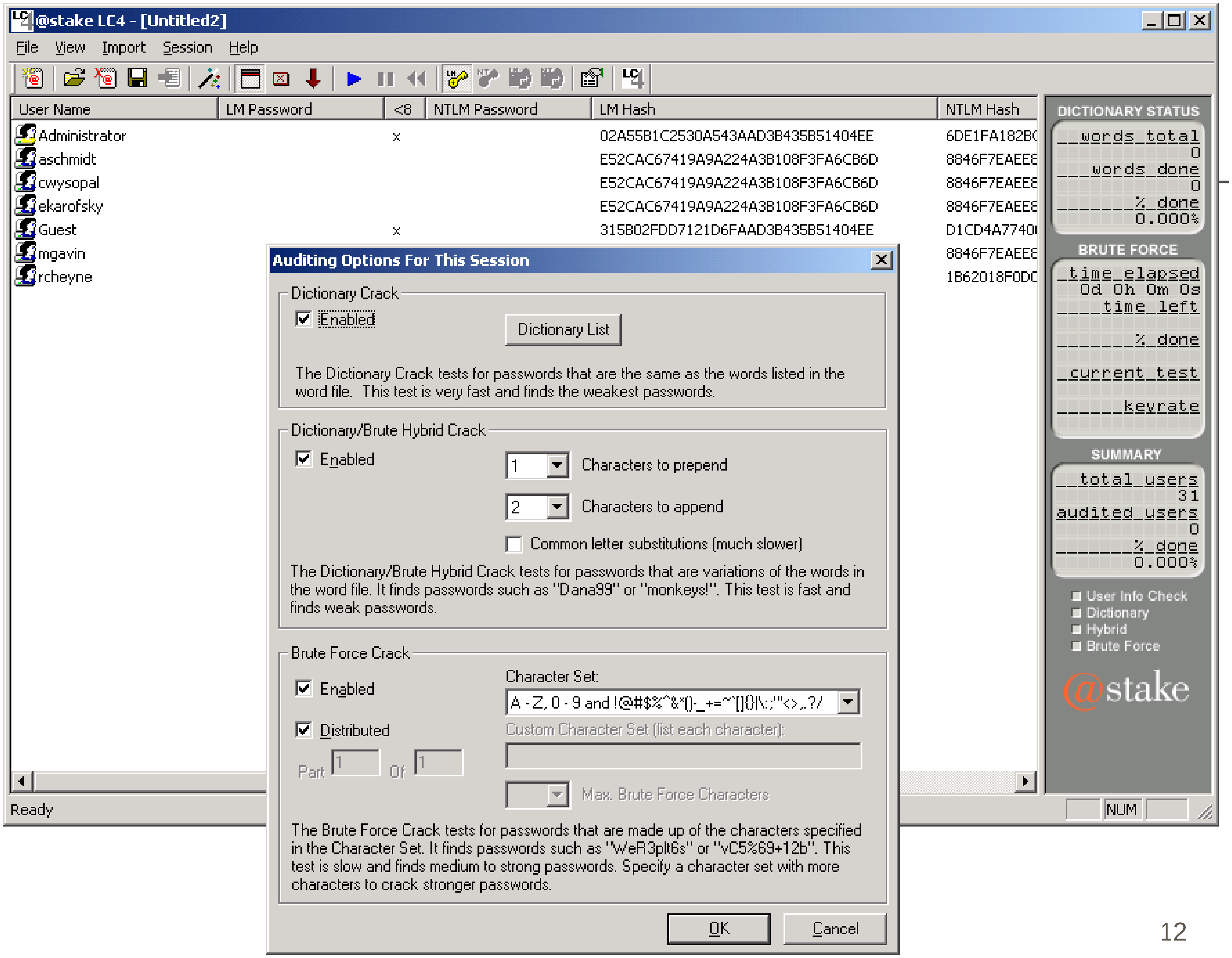
- Sniff the network for passwords being transmitted
- From Administrator’s emergency repair disk
- From back-up directory

Password Cracking on Windows

- L0phtCrack – lc4 (Windows)
  - Available at www.@stake.com/research/lc/
  - Password Auditing and Recovery Application
  - Default English dictionary 50,000 words
  - Does “hybrid” attacks
  - Our free trial version does not allow brute force (for $350 can purchase with that capability)
  - Works on weaker LanMan (LM) as well as NT hashes
  - Can sniff a network for LanMan hashed passwords
  - Can download from a local machine or remote computer the hashed password file

L0phtCrack (lc4)

- Some statistics (from the website)
  - L0phtCrack obtained 18% of the passwords in 10 minutes
  - 90% of the passwords were recovered within 48 hours on a Pentium II/300
  - The Administrator and most Domain Admin passwords were cracked



User Name	LM Password	<8	NTLM Password	LM Hash	NTLM Hash
Administrator		x		02A55B1C2530A543AAD3B435B51404EE	6DE1FA182B(
aschmidt				E52CAC67419A9A224A3B108F3FA6CB6D	8846F7EAEE8
cwysopal				E52CAC67419A9A224A3B108F3FA6CB6D	8846F7EAEE8
ekarofsky				E52CAC67419A9A224A3B108F3FA6CB6D	8846F7EAEE8
Guest		x		315B02FDD7121D6FAAD3B435B51404EE	D1CD4A7740
mgavin					8846F7EAEE8
rcheyne					1B62018F0DC

DICTIONARY STATUS: words total 0; words done 0; % done 0.000%

BRUTE FORCE: time elapsed 0d 0h 0m 0s; time left; % done; current test; keyrate

SUMMARY: total users 31; audited users 0; % done 0.000%

User Info Check
Dictionary
Hybrid
Brute Force

Auditing Options For This Session

Dictionary Crack

[✓] Enabled

The Dictionary Crack tests for passwords that are the same as the words listed in the word file. This test is very fast and finds the weakest passwords.

Dictionary/Brute Hybrid Crack

[✓] Enabled

1 Characters to prepend

2 Characters to append

[ ] Common letter substitutions (much slower)

The Dictionary/Brute Hybrid Crack tests for passwords that are variations of the words in the word file. It finds passwords such as "Dana99" or "monkeys!". This test is fast and finds weak passwords.

Brute Force Crack

[✓] Enabled

[✓] Distributed

Part 1 Of 1

Character Set: A - Z, 0 - 9 and !@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/

Custom Character Set (list each character):

Max. Brute Force Characters

The Brute Force Crack tests for passwords that are made up of the characters specified in the Character Set. It finds passwords such as "WeR3pIt6s" or "vC5%69+12b". This test is slow and finds medium to strong passwords. Specify a character set with more characters to crack stronger passwords.

Password Cracking on UNIX

- John the Ripper
- Available at http://www.openwall.com/john/
- Supports six hashing schemes including XP
- Old Unix used /etc/passwd to store passwords
- Password is stored after cryptographically altered
- Various algorithms (hash/encrypted) used by various Unix platforms
- /etc/password is readable by everyone
- Some Unix store in a shadow password file thus /etc/passwd does not contain the passwords since they are instead in /etc/shadow or /etc/secure, only root can access these files
- If shadow file used, must have root to copy

Password retrieval on Linux

- List of login names and usernames in /etc/passwd
- List of encrypted passwords in /etc/shadow
- Only /etc/shadow is enough to crack the passwords.
- Having both files makes it easier

John the Ripper

- Combine information from /etc/passwd and /etc/shadow into one file
- Use this file as input for John the Ripper
- John can create guesses by
  - Using built-in dictionary
  - Using account information
  - Using brute-force guessing algorithm

John the Ripper

- Scrambling used for each guess
- When a password is cracked, result displayed on screen
- During execution of this tool, hitting any key will give current guess and status
- Password complexity determines time needed for cracking them

Defenses against Password Cracking

- Select good passwords (not dictionary based)
- Change regularly
- Use tools to prevent easy passwords
- Use password cracking tests against own systems
- Protect system back ups that have password files
- Unix: activate password shadowing
- Windows: disable weaker LM authentication if no windows 95/98 machines on network

Agenda

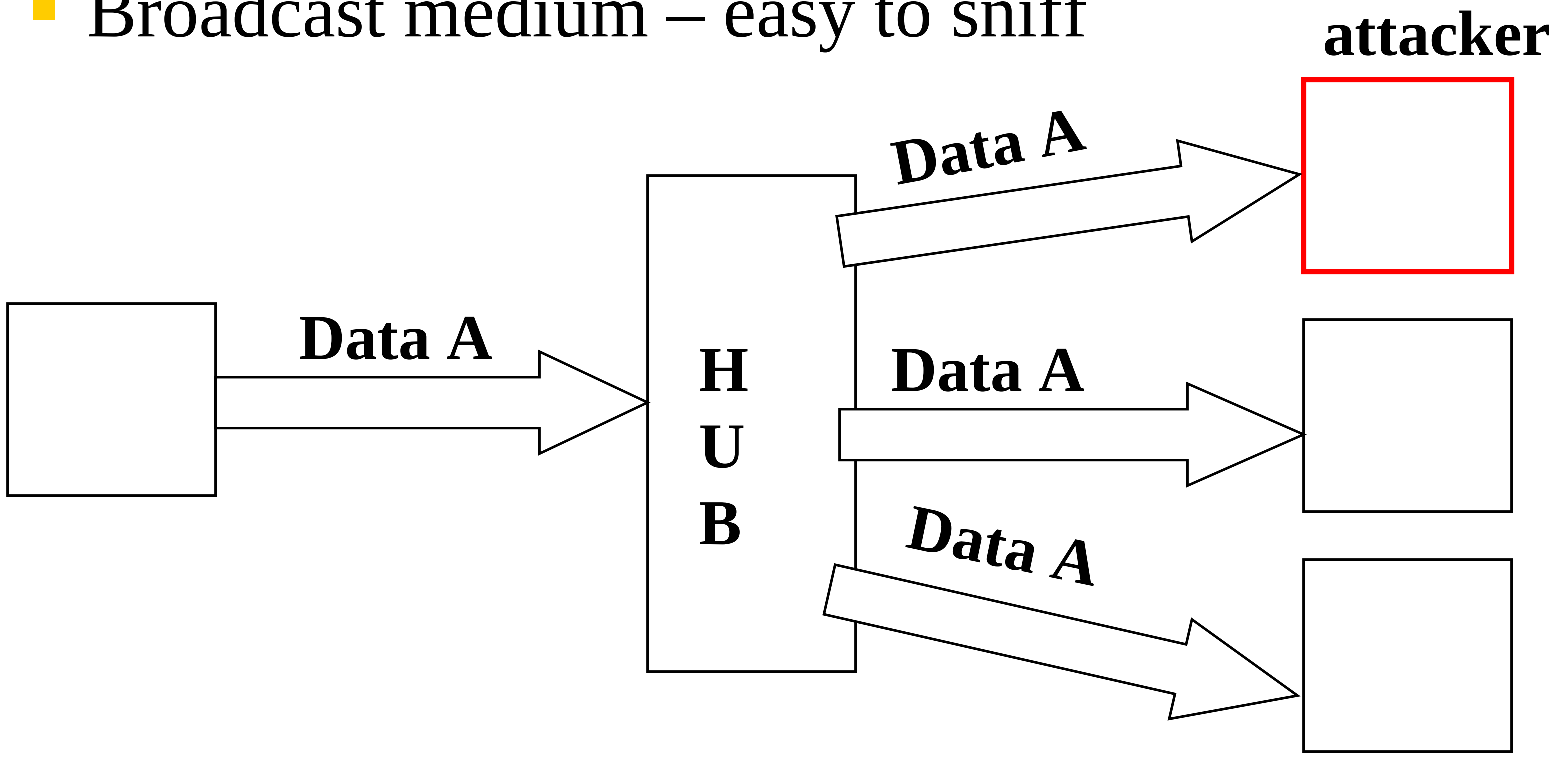
- ✓ Storing Passwords on the system
- ✓ Password Cracking on Windows and Linux
- ✓ Defenses against Password cracking
- **<u>Sniffing</u>**
- Defenses against Sniffing

Sniffing

- Collect information being transmitted on the network
- Attacker must be either on source, destination or intermediate network
- Sniffed information can be stored/logged

Sniffing traditional LANS

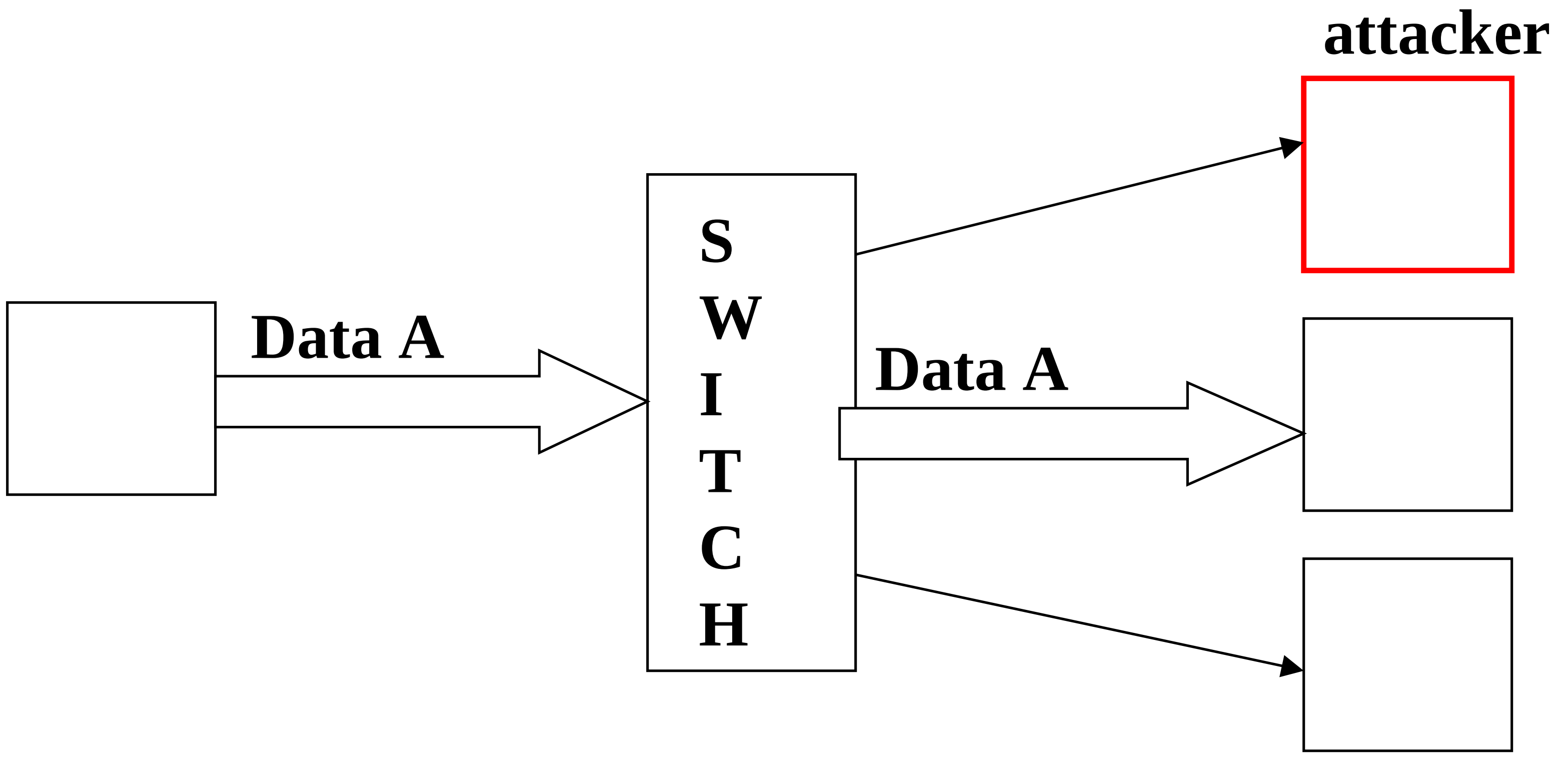
- Traditional networks
  - Broadcast medium – easy to sniff



Sniffing Switched LANS

- Switched LANS
  - Difficult to do, but possible
  - ARP Cache Poisoning - Attacker must inject packets into the network to redirect traffic
  - Attacker lies about the MAC address intercepts traffic
    - ARP tells which MAC address corresponds to which IP address

Sniffing Switched LANS



Sniffit

- Easy to use sniffer
- Available at: http://reptile.rug.ac.be/~coder/sniffit/sniffit.html
- Can be run in interactive mode
- Can be used to sniff traditional LANS
- For Switched LANS, must be used with ARP Cache Poisoning tools

Sniffit

- Conditions to use (from the Sniffit web page):
  - You should be ROOT on your machine
  - The machine has to be connected to a network
  - You have to be allowed to sniff (ethical condition)

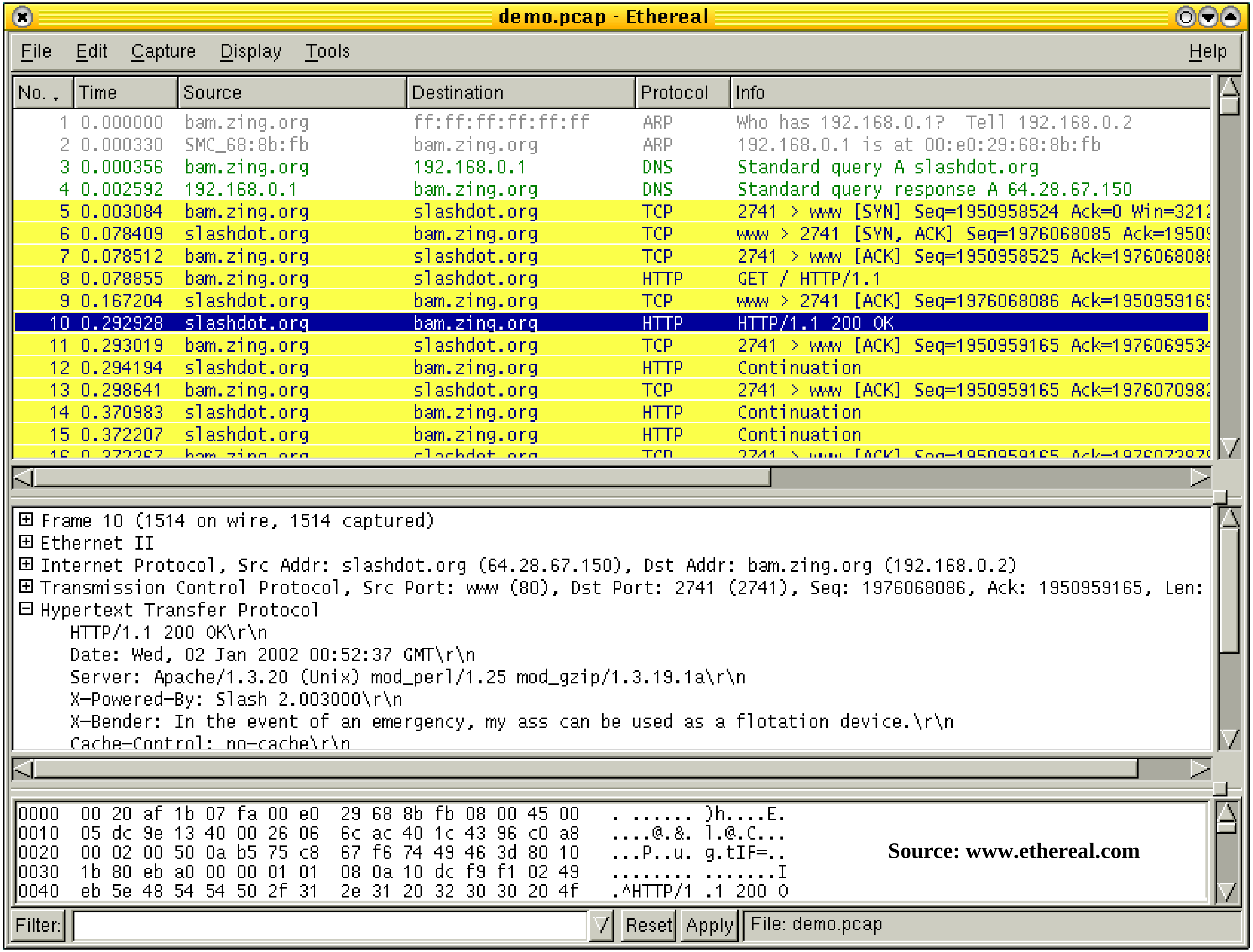
Sniffit – Interactive mode

- All TCP traffic can be viewed in main screen
- Traffic from each system and port to each system and port can be seen
- Has option to see data in a particular stream flow

ethereal

- From http://www.ethereal.com/
- Ethereal is a free network protocol analyzer for Unix and Windows.
- It allows you to examine data from a live network or from a capture file on disk.
- You can interactively browse the capture data, viewing summary and detail information for each packet.
- Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session.

Source: www.ethereal.com



```
No.  Time      Source         Destination        Protocol  Info
  1 0.000000 bam.zing.org   ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.0.1?  Tell 192.168.0.2
  2 0.000330 SMC_68:8b:fb   bam.zing.org       ARP       192.168.0.1 is at 00:e0:29:68:8b:fb
  3 0.000356 bam.zing.org   192.168.0.1        DNS       Standard query A slashdot.org
  4 0.002592 192.168.0.1    bam.zing.org       DNS       Standard query response A 64.28.67.150
  5 0.003084 bam.zing.org   slashdot.org       TCP       2741 > www [SYN] Seq=1950958524 Ack=0 Win=3212
  6 0.078409 slashdot.org   bam.zing.org       TCP       www > 2741 [SYN, ACK] Seq=1976068085 Ack=19509
  7 0.078512 bam.zing.org   slashdot.org       TCP       2741 > www [ACK] Seq=1950958525 Ack=197606808
  8 0.078855 bam.zing.org   slashdot.org       HTTP      GET / HTTP/1.1
  9 0.167204 slashdot.org   bam.zing.org       TCP       www > 2741 [ACK] Seq=1976068086 Ack=195095916
 10 0.292928 slashdot.org   bam.zing.org       HTTP      HTTP/1.1 200 OK
 11 0.293019 bam.zing.org   slashdot.org       TCP       2741 > www [ACK] Seq=1950959165 Ack=197606953
 12 0.294194 slashdot.org   bam.zing.org       HTTP      Continuation
 13 0.298641 bam.zing.org   slashdot.org       TCP       2741 > www [ACK] Seq=1950959165 Ack=197607098
 14 0.370983 slashdot.org   bam.zing.org       HTTP      Continuation
 15 0.372207 slashdot.org   bam.zing.org       HTTP      Continuation
 16 0.372267 bam.zing.org   slashdot.org       TCP       2741 > www [ACK] Seq=1950959165 Ack=197607287
```

```
⊞ Frame 10 (1514 on wire, 1514 captured)
⊞ Ethernet II
⊞ Internet Protocol, Src Addr: slashdot.org (64.28.67.150), Dst Addr: bam.zing.org (192.168.0.2)
⊞ Transmission Control Protocol, Src Port: www (80), Dst Port: 2741 (2741), Seq: 1976068086, Ack: 1950959165, Len:
⊟ Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Wed, 02 Jan 2002 00:52:37 GMT\r\n
    Server: Apache/1.3.20 (Unix) mod_perl/1.25 mod_gzip/1.3.19.1a\r\n
    X-Powered-By: Slash 2.003000\r\n
    X-Bender: In the event of an emergency, my ass can be used as a flotation device.\r\n
    Cache-Control: no-cache\r\n
```

```
0000  00 20 af 1b 07 fa 00 e0  29 68 8b fb 08 00 45 00   . ...... )h....E.
0010  05 dc 9e 13 40 00 26 06  6c ac 40 1c 43 96 c0 a8   ....@.&. l.@.C...
0020  00 02 00 50 0a b5 75 c8  67 f6 74 49 46 3d 80 10   ...P..u. g.tIF=..
0030  1b 80 eb a0 00 00 01 01  08 0a 10 dc f9 f1 02 49   ........ .......I
0040  eb 5e 48 54 54 50 2f 31  2e 31 20 32 30 30 20 4f   .^HTTP/1 .1 200 O
```

Source: www.ethereal.com

Defense against Sniffing

- Transmit encrypted data across a network
- Don’t use telnet, rsh,rlogin
- Use Secure Shell
- Use VPNs to encrypt data between systems
- Use switches instead of hubs – makes sniffing more difficult

Defense against Sniffing

- For critical systems
  - MAC level filtering on switches
  - Restrict MAC addresses that can send and receive data on specific switch plugs
  - Hard code ARP tables on critical systems