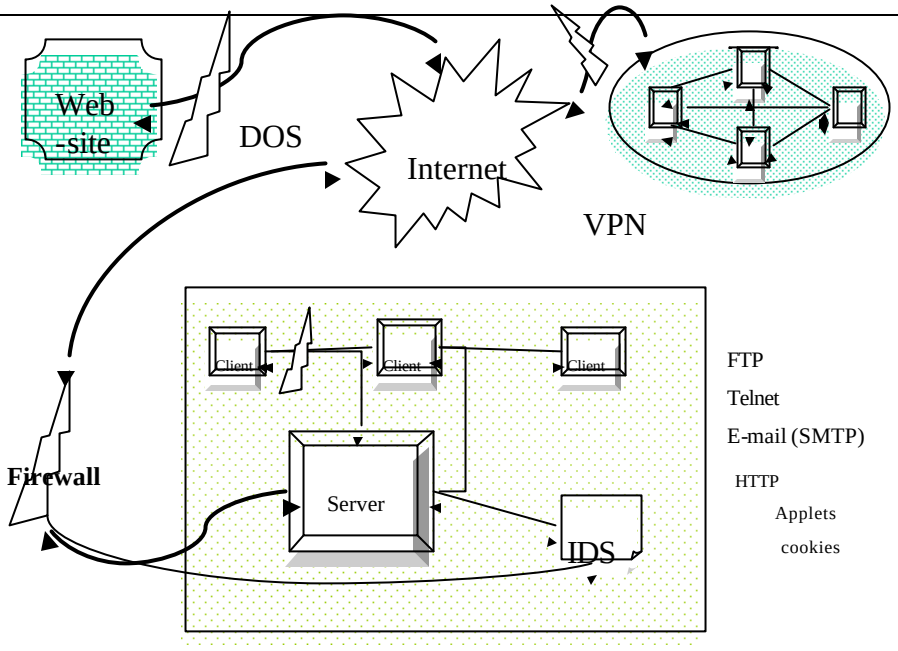
Intrusion Detection Systems

1



FTP
Telnet
E-mail (SMTP)
HTTP
Applets
cookies

2

WWW Challenges for Firewalls:

http= Hyper Text Transfer Protocol (TCP-based service):

- very flexible can be used for communications to other internet protocol such as SMTP, FTP, Gopher etc… Http can use any reliable transport.
- But this flexibility makes web-server difficult to secure
- http communications take place typically through port 80, but other ports can be used (>1023)

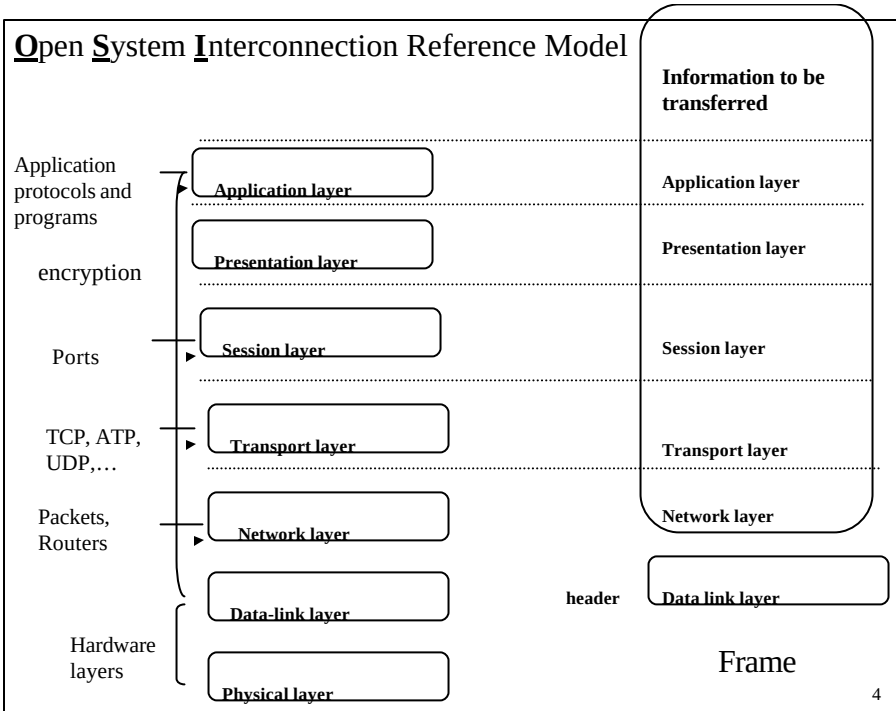
Extensions like CGI (Common Gateway Interface) scripts can generate vulnerabilities.

CGI's can have search engine for local search. Hackers can modify them to widen the search.

Internet Server Application Programming Interface (ISAPI; example: IIS)

3

Open **S**ystem **I**nterconnection Reference Model



4

Level	Name	IP Protocols
5	Session / Presentation / Application	TFTP, Bootp, SMTP, NNTP, POP, RPC, Portmapper, HTTP, Telnet, FTP, DHCP, Socks, MOUNT, NFS, NLM, NSM, PMAP
4	Transport	TCP, UDP
3	Network	IP, ICMP, ARP, RARP
2	Data link	Not currently used
1	Physical	Ethernet packets

5

Port Numbers

from Internet Assigned Numbers Authority (IANA). IANA maintains the Assigned Numbers RFC.

- The port numbers are divided into three ranges:
  - **The Well Known Ports are those from 0 through 1023.**
  - **The Registered Ports are those from 1024 through 49151**
  - **The Dynamic and/or Private Ports are those from 49152 through 65535**
- smtp 25/tcp Simple Mail Transfer
- smtp 25/udp Simple Mail Transfer
- http 80/tcp World Wide Web HTTP
- http 80/udp World Wide Web HTTP
- www-http 80/tcp World Wide Web HTTP
- www-http 80/udp World Wide Web HTTP

- netbios-ssn 139/tcp NETBIOS Session Service
- netbios-ssn 139/udp NETBIOS Session Service
- ias-auth 2139/tcp IAS-AUTH
- ias-auth 2139/udp IAS-AUTH

6

Snort - Lightweight Intrusion Detection for Networks

7

Introducing Snort

- Snort is:
  - Small (~110K source distribution)
  - Portable (Linux, Solaris, *BSD, IRIX, HP-UX)
  - Fast (High probability of detection for a given attack on “average” networks)
  - Configurable (Easy rules language, many reporting/logging options)
  - Free (GPL/Open Source Software)

8

Snort Design

- Packet sniffing network intrusion detection system
- Libpcap-based sniffing interface
- Rules-based detection engine
- Multiple output options
  - decoded logs, tcpdump formatted logs
  - real-time alerting to syslog, file, winpopup

9

Snort is a libpcap-based packet sniffer and logger that can

be used as a lightweight network intrusion detection system (NIDS).

It features rules based logging to perform content pattern matching and detect

a variety of attacks and probes, such as buffer overflows, stealth port

scans, CGI attacks, SMB probes, and much more.

Snort has real-time alerting capability, with alerts being sent to syslog, Server Message Block (SMB), "WinPopup" messages, or a separate "alert" file.

When the IIS Showcode web exploits were revealed on the Bugtraq mailing list, Snort rules to detect the probes were available within a few hours.

10

Rules Format

```
alert tcp any any -> 192.168.1.0/24 143 (content: "|90C8 C0FF FFFF|/bin/sh"; msg:
"IMAP buffer overflow!";)
```

- Two sections to a rule
  - rule header: **alert tcp any any -> 192.168.1.0/24 143**
  - rule options:

    (content: "|90C8 C0FF FFFF|/bin/sh"; msg: "IMAP buffer overflow!";)

Current version of Snort (1.3.1) has fourteen rule options available

11

--------syslog_test_1.c----------

```
#include
char buffer[4028];
void main()
        {
          int i;
          for (i=0; i<=4028; i++)
        buffer[i]='A';
         syslog(LOG_ERR, buffer);
        }
```

--------end syslog_test_1.c----------

12

```
bash$ gcc -g buf.c -o buf
bash$ buf
Segmentation fault (core dumped)
```

```
bash$ gdb buf
(gdb) run
Starting program: /usr2/home/syslog/buf
Program received signal 11, Segmentation fault
0x1273 in vsyslog (0x41414141, 0x41414141, 0x41414141, 0x41414141)
```

The 41's you see are the hex equivalent for the ascii character 'A'

13

Rule options form the heart of Snort's intrusion detection engine:

```
alert tcp any any -> 192.168.1.0/24 143 (content: "|90C8 C0FF FFFF|/bin/sh"; msg: "IMAP buffer overflow!";)
```

All Snort rule options are separated from each other using the semicolon ";" character.

Rule option keywords are separated from their arguments with a colon ":" character.

```
alert tcp any any -> 192.168.1.0/24 80 (content: "cgi-bin/phf"; depth: 22; msg:"CGI-PHF access";)
```

If you're searching for "cgi-bin/phf" in a web-bound packet, you probably don't need to waste time searching the payload beyond the first 20 bytes!

14

msg - prints a message in alerts and packet logs

logto - log the packet to a user specified filename instead of the standard output file

ttl - test the IP header's TTL field value

tos - test the IP header's TOS field value

id - test the IP header's fragment ID field for a specific value

ipoption - watch the IP option fields for specific codes

fragbits - test the fragmentation bits of the IP header

dsize - test the packet's payload size against a value

flags - test the TCP flags for certain values

seq - test the TCP sequence number field for a specific value

ack - test the TCP acknowledgement field for a specific value

itype - test the ICMP type field against a specific value

icode - test the ICMP code field against a specific value

icmp_id - test the ICMP ECHO ID field against a specific value

icmp_seq - test the ICMP ECHO sequence number against a specific value

15

content - search for a pattern in the packet's payload

content-list - search for a set of patterns in the packet's payload

offset - modifier for the content option, sets the offset to begin attempting a pattern match

depth - modifier for the content option, sets the maximum search depth for a pattern match attempt

nocase - match the preceeding content string with case insensitivity

session - dumps the application layer information for a given session

rpc - watch RPC services for specific application/proceedure calls

resp - active response (knock down connections, etc)

react - active response (block web sites)

16

Honeypot Monitor

- Honeypots are “deception systems” that perform intrusion detection by inclusion
  - Gets rid of all the false alarms!
- Use Snort’s filtering capability to log only the traffic (but all the traffic!) going to the honeypot
- Post process the data with a good ruleset

17

Scan Detection/Traps

- Snort has no formal port scan detection mechanism
- Setup rules to log traffic to known closed ports & unused addresses
- Poor man’s honeypot/port scan detector

Alert tcp any any -> 10.1.1.0/24 100:600 (flags: S; msg: “TRAP!”;)

18

Other Fun Stuff

- Snort is a packet sniffer, can be used to analyze traffic in real-time
- Motivated people can write rules to pick up all sorts of naughty things
  - SQL/ODBC, ActiveX, Java/JavaScript, Macro Viruses

19

How to Add Sniffing Capabilities into your Application

(Knowledge of C or C++ programming in the Microsoft Win32 environment is assumed)

To start capturing network traffic, you must first connect to the network adapter whose traffic you want to monitor.

A computer can have an Ethernet card, ISDN adapter and Dial-Up Adapter installed. All three of them should have a network adapter driver installed.

You first need to retrieve the proper name for the adapter. This is done by calling the NmGetAdapter function.

20

```
char szName[32], szDesc[128];

        rc = NmGetAdapter(hConn, NM_GET_FIRST, szName, szDesc);

        while (rc == TRUE) {

        // szName contains the adapter name as must be specified

        // in the NmBindAdapter call. szDesc has an adapter description.

        rc = NmGetAdapter(hConn, NM_GET_NEXT, szName, szDesc);

        }
```

21

or use:

```
        rc = NmGetAdapter(hConn, NM_GET_FIRST, szName, szDesc);

        if (rc == TRUE)

        NmGetAdapter(hConn, NM_GET_CLOSE, NULL, NULL);
```

to get the first available adapter name.

Once this is done you are ready to bind to the adapter using the connection handle and the adapter name. If binding succeeds, network packet that pass the adapter will be delivered to the callback function. (Window procedure)

22

Care should be taken to process the incoming packets as quickly as possible.

The sniffer driver has an internal packet buffer of limited size. If this buffer is full, new incoming packets will be discarded until free packet slots become available.

Since network traffic can be very high because the sniffer is listening in to network traffic for all computers on the network segment, not just for traffic from or to our own computer, the callback function should try to make the processing time as short as possible.

You can check for missed packets by looking at the dwCount parameter.

23

Distinct Network Monitor offers two different modes that you can use to capture your network traffic. You can either capture it to a disk file for later analysis or capture it directly to the window on your screen.

When capturing to a disk file, the packets are saved directly to a disk file without displaying them on the screen.

Capturing to screen displays the packets as they come along, they are not saved on disk.

http://www.distinct.com/monitor

Forensic analysis

24

Advanced Techniques *cont'd*

- **Computational immunology**
  - based on biological analogies (e.g., self vs. non-self discrimination)
  - build up a database of observed short sequences of system calls for a program and detect when the observed program behavior exhibits short sequences not in that database (U. of NM)
  - allows the detection of tampered or malicious programs or other suspicious events
  - this potentially lightweight method is being implemented in small, autonomous agents in a CORBA environment (ORA)

parc | X

25

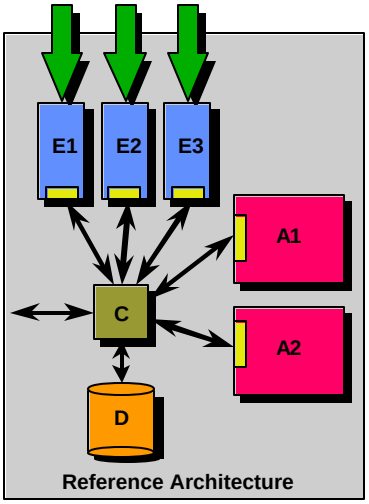
Advanced Techniques *cont'd*

- **Automated response** (Boeing)
  - Integrates firewall, intrusion detection, filtering router, and network management technologies
  - Local intrusion detectors determines threat presence
  - Firewalls communicate intrusion detection information to each other
  - Firewalls cooperate to locate the intruder
  - Network managers automatically reconfigure the network to thwart the attack
  - Firewalls and filtering routers dynamically alter filtering rules to block the intruder
  - Dynamic reconfiguration of logging, monitoring, and access control in response to detected suspicious activity
  - "Fusion" of intrusion-detection data reported by different detectors
  - The monitoring is also adapted as part of the response, to help pinpoint the problem and its source

parc | X

26

Common Intrusion Detection Framework



- **Standard Interfaces**
  - an interconnection framework for data collection, analysis, and response components
  - extensible architecture
  - reuse of core technology
  - facilitate tech transfer
  - reduce cost

Standard API
E Event Generator
A Event Analyzer
D Event Database
C System-specific Controller

parc | X

Intrusion Detection: New Directions

Teresa Lunt
Xerox Palo Alto Research Center
tlunt@parc.xerox.com

parc | X

Conclusions

- Currently available technology is not adequate for the problem
- Promising methods under investigation show significant improvement over current technology
- There is still a lot more to be done

parc | X