

Introduction to Network Security

Chapter 8

Application Layer Overview

Dr. Doug Jacobson - Introduction to
Network Security - 2009

1

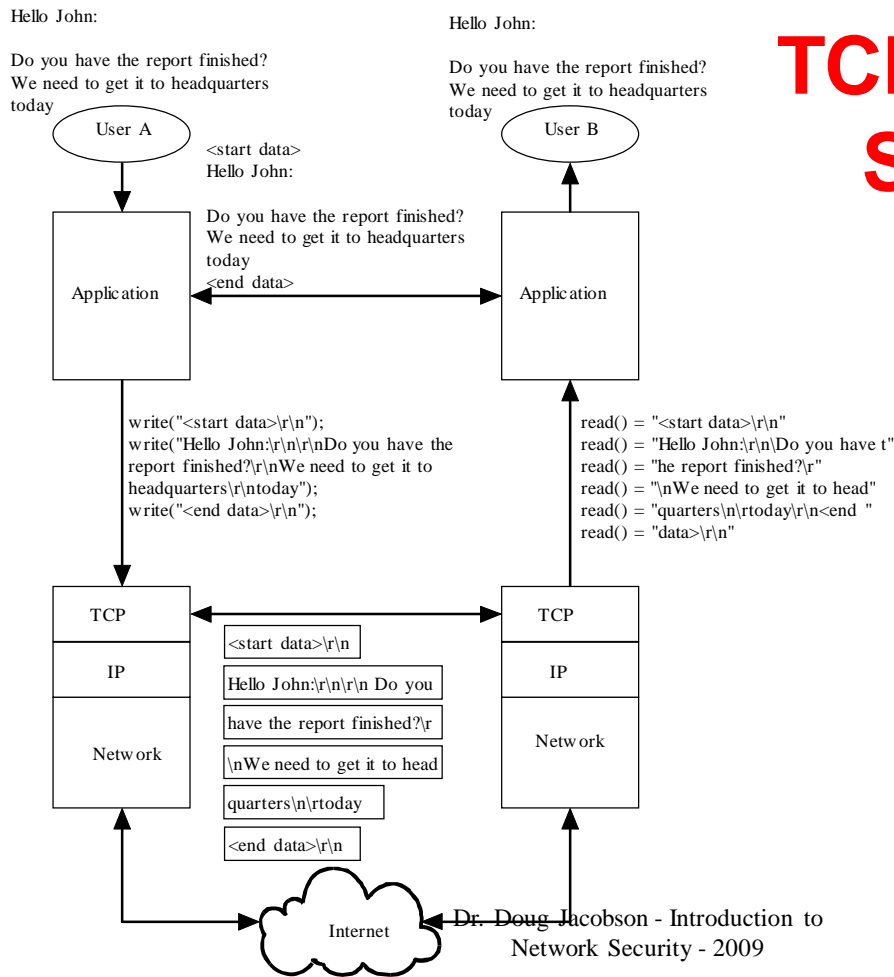
Application Layer Security

- TCP stream Service
- Socket Layer
- Common Attack Methods

Dr. Doug Jacobson - Introduction to
Network Security - 2009

2

TCP Stream Service



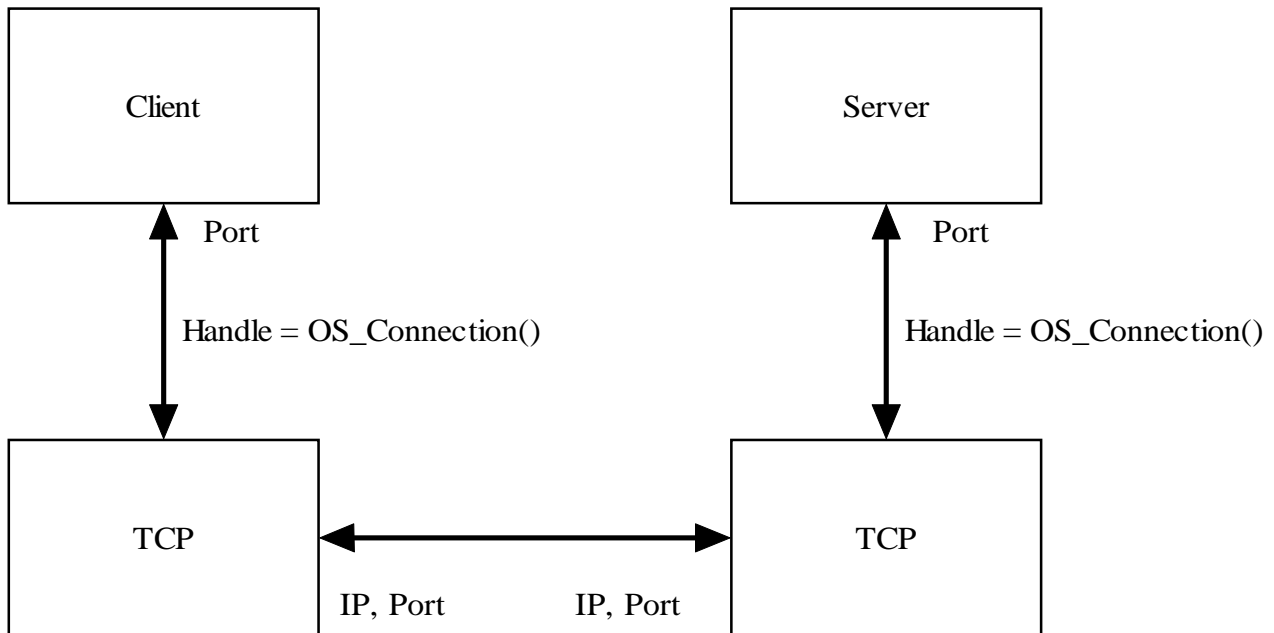
Dr. Doug Jacobson - Introduction to Network Security - 2009

3

Sockets

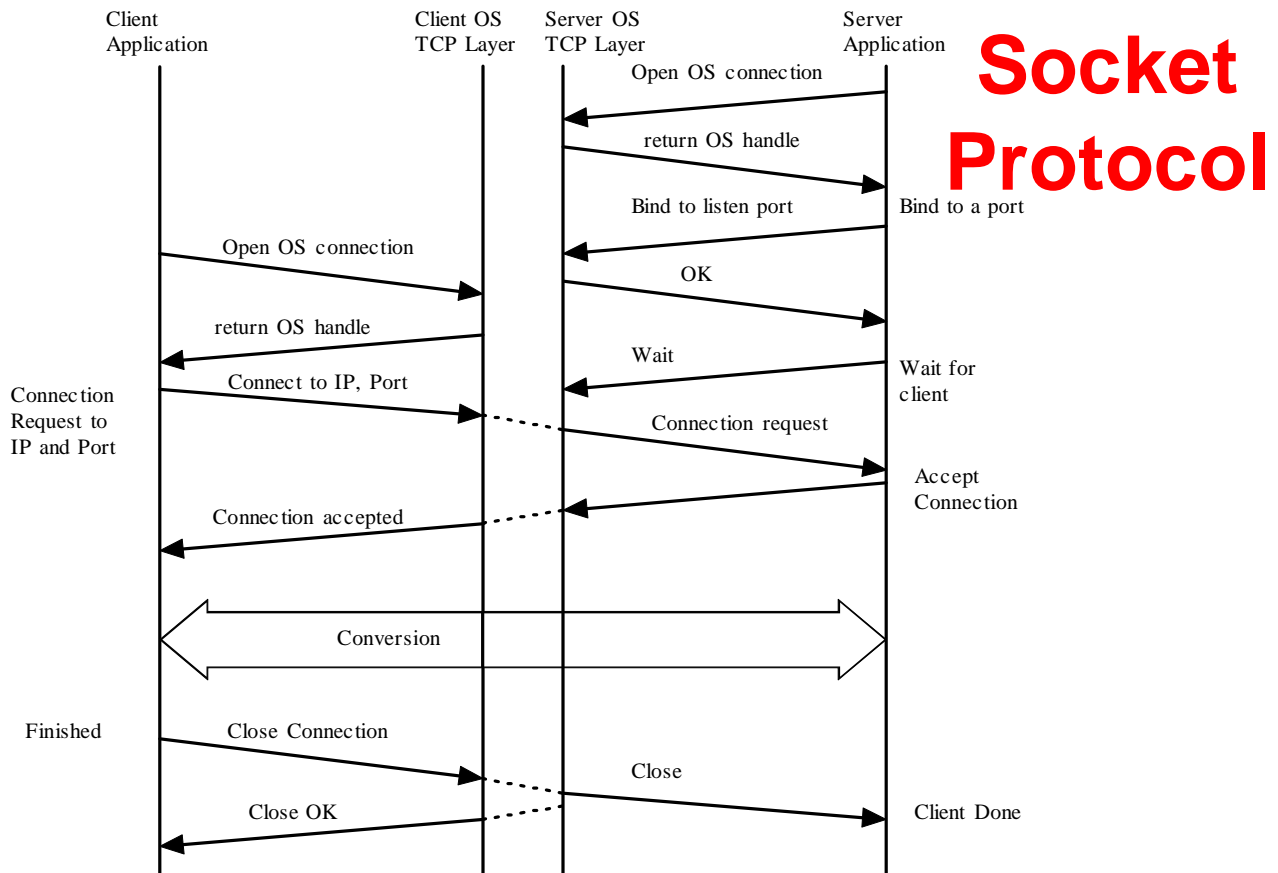
- Application to TCP interface
- Socket protocol

Application / TCP Interface



Dr. Doug Jacobson - Introduction to Network Security - 2009

5



Dr. Doug Jacobson - Introduction to Network Security - 2009

6

Socket Code – Server Side

```
nsaddr.sin_family = AF_INET;
nsaddr.sin_addr.s_addr = INADDR_ANY; // Accept connection from all
/* nsaddr.sin_addr.s_addr = inet_addr("129.186.5.101"); */
nsaddr.sin_port = htons(2000);
// Open stream port.
if ((vs = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
    printf("socket(SOCK_DGRAM): %d\n",errno);
    exit(1);
}
// bind stream to port 2000 from any address
if (bind(vs, (struct sockaddr *)&nsaddr, sizeof(nsaddr)) < 0) {
    printf("bind(vs, %s[%d]) errno = %d\n "
        ,inet_ntoa(nsaddr.sin_addr), ntohs(nsaddr.sin_port),errno);
    perror("bind error");
    exit(1);
}
fprintf(stderr,"SERVER: bind(vs, %s[%d]):\n ",
    inet_ntoa(nsaddr.sin_addr), ntohs(nsaddr.sin_port));
```

Dr. Doug Jacobson - Introduction to
Network Security - 2009

7

Socket Code – Server Side

```
printf("SERVER: listen waiting\n");
// allow 5 pending connection requests to this port
if ((listen(vs,5)) < 0) {
    perror("listen");
    exit(1);
}
printf("SERVER: waiting buf size = %d\n",sizeof(buf));
from_len = sizeof(from_addr);
// wait for incoming connection
if ((ns = accept(vs, &from_addr, &from_len)) < 0) perror("accept");
```

Dr. Doug Jacobson - Introduction to
Network Security - 2009

8

Socket Code – Client Side

```
// this calls the DNS system
h_name = gethostbyname("vulcan.ee.iastate.edu");
/* s_name = getservbyname("phone", "udp");*/
/* sin.sin_port      = s_name->s_port; */
sin.sin_family = AF_INET;
sin.sin_port = htons(2000); // port to connect to
sin.sin_addr.s_addr = *(u_long *)h_name->h_addr;
printf("port = %d %s\n",ntohs(sin.sin_port),
inet_ntoa(sin.sin_addr));
// open socket
sockFD = socket(AF_INET, SOCK_STREAM, 0);
// open connection to server
if (connect(sockFD, &sin, sizeof(sin)) < 0) {
    perror("connect request");
    (void) close(sockFD);
    exit(1);
}
```

Dr. Doug Jacobson - Introduction to
Network Security - 2009

9

Socket Code – data xfer

```
strcpy(buf,"from client");
// client sends first
if (send(sockFD, buf, strlen(buf),0) != strlen(buf)) {
    perror("send request");
    (void) close(sockFD);
    exit(1);
}
// Client waits for answer
cp = answer;
if ((n = recv(sockFD, cp, 100, 0)) < 0){
    perror("SendRequest");
    (void) close(sockFD);
}
cp[n] = 0;
printf("===<%s>===\n",cp);
(void) close(sockFD);
```

Dr. Doug Jacobson - Introduction to
Network Security - 2009

10

Socket Code – data xfer

```
printf("SERVER: accepted call\n");
// print where the connection came from
fprintf(stderr,"SERVER: from_addr(ns, %s[%d]):\n ",
        inet_ntoa(from_addr.sin_addr), ntohs(from_addr.sin_port));
// get the data from the client
blen = recv(ns,buf,sizeof(buf), 0);
buf[blen] = 0;
printf("SERVER: --<%s>--\n",buf);
strcpy(buf,"hello");
printf("SERVER: sending\n");
// send response to client
if (send(ns, buf, strlen(buf), 0) != strlen(buf)) {
    perror("Sendto");
}
// shutdown connection, leaves socket open
shutdown(ns,2);
```

Dr. Doug Jacobson - Introduction to
Network Security - 2009

11

Application Layer Vulnerabilities

- Same four categories
- Applications do have some attacks in common
- Attacks are often limited to the application
- Can allow access to the computer (privileged applications are a common target)

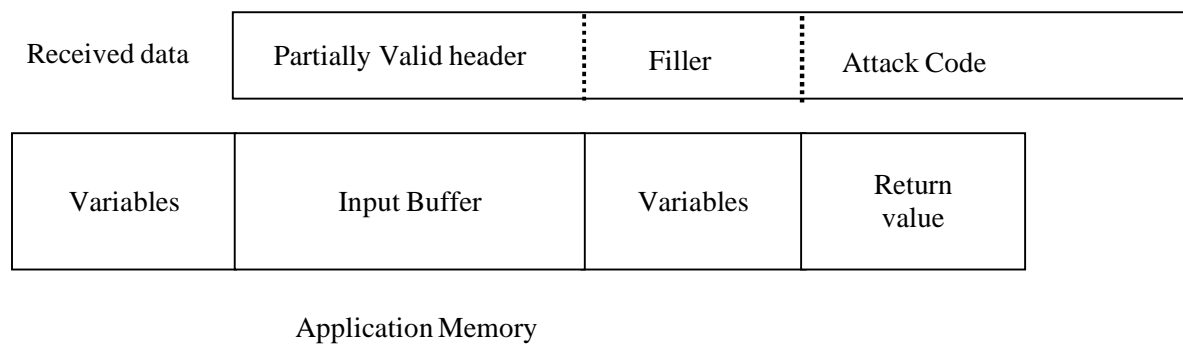
Dr. Doug Jacobson - Introduction to
Network Security - 2009

12

Header-Based

- Common attack against applications
- Most applications have a freeform header which means the header must be parsed
- Buffer overflow is a common form

Buffer overflow



Protocol-Based

- Application specific
- Often part of an authentication attack

Authentication-Based

- The most common type of attack
- Two types
- Direct Attack
 - Using the applications authentication mechanism to gain access (password guessing)
- Indirect Attack
 - Using one of the other attack categories to circumvent authentication (primary focus)

Traffic-Based

- DOS
- Sniffing