Vorlesung Systemnahe Programmierung – WS 2015/16

Prof. Dr. Matthew Smith, Dr. Matthias Frank, Sergej Dechand

Bonus-Zettel

Ausgabe: 30.11.2015; Abgabe bis zum 17.01.2016, 23:59 Uhr; Anmeldung zum Bonus-Zettel bis zum 03.01.2016, 23:59 Uhr.
Die Vorführung erfolgt nach Absprache vom 18.-29.01.2016

Die Bearbeitung des Bonus-Zettels ist freiwillig. Sie können auch ohne dessen Bearbeitung alle benötigen Punkte auf den Übungszetteln sammeln. Da die Besprechung nicht im Rahmen der regulären Vorführungen erfolgt, müssen Sie sich zu dem Übungszettel anmelden. Schicken Sie dazu eine Mail mit Ihrem und dem Namen Ihres Abgabepartners bis zum 03.01.2016, 23:59 Uhr an

raphael.ernst@fkie.fraunhofer.de.

Verwenden Sie als Betreff unbedingt „SysProg Bonus-Zettel".

Schicken Sie bitte alle Fragen, die den Bonus-Zettel betreffen auf die bekannte Mailingliste zur Vorlesung.

Schicken Sie die Lösung Ihres Übungszettels an

raphael.ernst@fkie.fraunhofer.de.

Verwenden Sie als Betreff unbedingt „SysProg Bonus-Zettel - Lösung". Sollten Sie innerhalb der Bearbeitungszeit mehr als eine Lösung einreichen, wird nur die letzte Version bewertet.

Alle Programme müssen mit einem Makefile abgegeben werden. Mit dem Aufruf „make install" soll Ihr Programm und ggf. benötigte Abhängigkeiten installiert werden. Mit „make start" soll Ihr Programm gestartet werden.
Als Hostsystem kommt für beide Aufgaben eine Ubuntu 14.04 Basisinstallation zum Einsatz. Sollten Sie Pakete benötigen, die bei der Basisinstallation nicht dabei sind, können Sie diese im Makefile installieren lassen. Nutzen Sie dazu apt-get.

Aufgabe A: TCP-Stack (36 Punkte)

Fast alle Betriebssysteme haben einen eigenen TCP-Stack wobei unterschiedliche Implementierungen realisiert wurden (z.B. New Reno, Reno, Tahoe, Vegas). Trotz der Unterschiede sind sie zueinander kompatibel. D.h. selbst sehr einfache TCP-Stacks können mit aktuellen Implementierungen Daten austauschen.
In dieser Aufgabe sollen Sie ihren eigenen TCP-Stack als C Userland Library in Linux implementieren und in ein Programm integrieren. Die API der Library muss folgende Funktionen bereitstellen:

- **int accept(uint16_t *port,* unit32_t *ipaddress*)**
  Beschreibung

accept() versucht einen TCP *port* für die *ipaddress* zu öffnen.
Rückgabewert
Ist der Aufruf erfolgreich wird ein nicht-negativer Wert zurückgegeben, der die Verbindung identifiziert. Dieser Wert wird File Descriptor genannt.
Im Fehlerfall wird -1 zurückgegeben.

- **ssize_t read(int fd, void* buf, size_t count)**
  Beschreibung
  read() liest bis zu *count* Bytes vom File Descriptor *fd* in the Puffer der bei Adresse *buf* beginnt.
  Rückgabewert
  Ist der Aufruf erfolgreich, wird die Anzahl der gelesenen Bytes zurückgegeben. Werden Null Bytes zurückgegeben wurde die Verbindung beendet. Es ist kein Fehler falls weniger als die angefragten Bytes gelesen wurden. Dies kann passieren wenn die Gegenseite weniger Bytes versendet hat als in den Puffer passen.
  Im Fehlerfall wird -1 zurückgegeben.
- **ssize_t write(int fd, void* buf, size_t count)**
  Beschreibung
  write() schreibt *count* Bytes in den File Descriptor *fd* aus dem Puffer der bei Adresse *buf* beginnt.
  Rückgabewert
  Ist der Aufruf erfolgreich, wird die Anzahl der gesendeten Bytes zurückgegeben.
  Im Fehlerfall wird -1 zurückgegeben.
- **void close(int fd)**
  Beschreibung
  close() schließt die Verbindung, die durch den File Descriptor *fd* beschrieben ist. Nach dem Aufwurf von close() darf die Funktion write() nicht länger aufgerufen werden. Read()-Aufrufe können so lange erfolgen, bis auch die Gegenseite die Verbindung beendet hat.
  Rückgabewert
  Keiner.

Implementieren Sie ein Programm, das ihre Library benutzt und die Payload der TCP-Verbindung in eine Datei schreibt. Beendet der Sender die Verbindung soll ihr Programm ebenfalls die Verbindung schließen und sich im Anschluss beenden.
Sie dürfen auf C-Libraries zurückgreifen, die Sie unterstützten. Libraries, die TCP-Stacks implementieren sind nicht zugelassen.

Ihr Programm muss nach dem Start folgende Informationen anzeigen:

- Die IP-Adresse für die Ihr Programm Daten entgegen nimmt. Verwenden Sie die übliche Notation in vier Zahlenblöcken (0-255), getrennt durch einen Punkt.
- Die Portnummer auf der Daten entgegen genommen werden.
- Der Dateiname in den die Daten gespeichert werden.
- Für alle eingehenden Datenpakte soll die Anzahl der Datenbytes angezeigt werden.

Ihr Programm muss unter Ubuntu 14.04 laufen.

Hinweise

- Suchen Sie nach „raw socket linux“ falls Sie einen Startpunkt zur Implementierung suchen!
- Nutzen Sie Tools wie tcpdump oder wireshark, um den Datenaustausch auf der

Leitung zu beobachten. Auf diese Weise können Sie Implementierungsfehler aufspüren.

- Testen Sie Ihre Implementierung! Als Gegenstelle können Sie netcat nutzen, um eine Datei zu übermitteln („cat /etc/hostname | nc 127.0.0.1 4711“ überträgt den Inhalt der Datei /etc/hostname an 127.0.0.1 auf Port 4711). Prüfen Sie, ob die empfangenen Daten mit den versendeten übereinstimmen.
- Ihr System hat höchst wahrscheinlich einen eigenen TCP-Stack. Damit Sie nicht mit diesem in Konflikt geraten, darf Ihr TCP-Stack die TCP/IP-Pakete für eine andere IP-Adresse empfangen und verarbeiten. Diese Adresse kann in ihrer accept()-Funktion mit dem Parameter *ipaddress* spezifiziert werden.
- Pseudo-Code einer Beispielanwendung, die Ihre Library nutzt:

```
int fd = accept(htons(4711), htons(3232235521)); // Listen on port 4711 and IP 192.168.0.1
const int buf_size = 1024;
char buf[buf_size];
while (read(fd,&buf,buf_size) > 0) {
 //write data to file
}
close(fd)
print("Connection terminated or error");
```

Aufgabe B: DNS-Traffic Umleitung (12 Punkte)

DNS Spoofing ist eine Angriffsart bei der ein Angreifer DNS Anfragen mit falschen IP Adressen beantwortet, um die Anfragen des Opfers umzuleiten. Schreiben Sie einen eigenen DNS Spoofer in C oder Go.

Ihr Spoofer muss seine Konfiguration aus einer Datei mit folgendem Format lesen:

<hostname 1>|<IP 1>
<hostname 2>|<IP 2>
…
<hostname N>|<IP N>

Alle Anfragen an einen Hostnamen, der in der Konfiguration hinterlegt ist, sollen mit der dazugehörigen IP Adresse beantwortet werden.