1 IP Tables

1.1 Initialisieren

```
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

1.2 NAT (Port Forwarding)

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 443 -j DNAT --to 192.168.0.x:443

-t = Tabelle
-A = Füge Regel zu ausgewählter Kette hinzu
PREROUTING: Bearbeiten der Pakete sobald sie reinkommen
POSTROUTING: Pakete erst bearbeiten, sobald sie rausgehen
-o = out-interface
-i = in-interface
-p = Protkoll
--dport = Destination Port
--sport = Source Port
eth0 = In diesem Fall Interface zum Internet
eth1 = In diesem Fall Interface ins Interne Netz
-j = Auszuführende Regel

1.3 Beispiel Forwarding Regel

iptables -A FORWARD -p tcp -i eth0 -d 192.168.101.x --dport 443 -j ACCEPT

iptables -A FORWARD -p tcp -o eth0 -s 192.168.101.x --sport 443 -j ACCEPT

-d = destination IP
-s = source IP

1.4 Beispiel Nur hergestellte Verbindung

iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT

-m = match
--state = der zu vergleichende Status

1.5 Beispiel DHCP-Server

iptables -A INPUT -i eth1 -p udp --dport 67 --sport 68 -j ACCEPT

iptables -A OUTPUT -o eth1 -p udp --dport 68 --sport 67 -j ACCEPT

--dport = destination Port
--sport = source Port

2 Funktionen & Bedenklichkeiten

Bedenklich	Attacke	Verbesserung
char *gets(char *str); gets(song)	Buffer Overflow	char *fgets(char *str, int num, FILE *stream); fgets(song, sizeof(song), stdin)
int sprintf(char *str, const char *fomat, …); int sprintf(command, „get %s.mp3“, song);	Buffer Overflow	int snprintf(char *s, size_t n, const char * format, …); int snprintf(command, sozeof(command), „get%s.mp3“, song); command[sizeof(command)-1]=‘\0‘;
size_t strlen(const char *s); len = strlen(str);	Buffer Overflow	size_t strnlen(const char *s, size_t maxlen); len = strnlen_s(str, sizeof str);

char * strcpy (char * destination, const char * source);
`strcpy (str2,str1);`
	Buffer Overflow	size_t strlcpy(char *destination, const char *source, size_t size); ODER
		char * strncpy(char *destination, const char *source, size_t size);
		strncpy (str2, str1, sizeof(str2));
		len = strlcpy(str2, str1, sizeof(str2));

char *strcat(char *destination, const char *source) strcat(to, from)	Command Injection	size_t strlcat(char *dst, const char *src, size_t size); strlcat(to, from, sizeof(to));
gets(input); [anz. in n= „%s%n\n“,buf,&n] printf(input);	Formatstring-Attacke	Herausfiltern der Zeichen. Erkennung → %n (anzahl Zeichen) [input = „%s%n\n“,buf,&n]
filename= mktemp(template); fd = open(filename, O_RDWR);	Race Condition	Zwischen dem erzeugen und dem Öffnen der Datei existiert eine Race Condition, da die Datei in der Zwischenzeit der beiden Aufrufe geändert worden sein könnte

3 Sonstige Bedenklichkeiten

- Kein Salz
- least privilege
- Rückgabewerte von Funktionen nicht ausgewertet
- sscanf oder ähnliches
- Signale nicht abgefangen
- Schutz vor Swapping fehlt
- Challenge kein Zufallswert