Contents

1 Standard Frage

1. Wo mit Beschäftigt sich IT-Sicherheit(Definition)?
IT-Sicherheit beschäftigt sich mit der Vorbeugung, dem Erkennen und der Reaktion auf Ereignisse, die die Integrität der Daten, die Nutzbarkeit der Systeme und die (digitale) Privatsphäre gefährden.

2. Vor welchen drei Gefährdungen müssen Rechner- und Netzwehrkomponenten geschützt werden?

- Spionage
- Sabotage
- Missbrauch

3. Nennen Sie die in der Vorlesung genannten drei Schutzziele für Rechner- und Netzkomponenten.

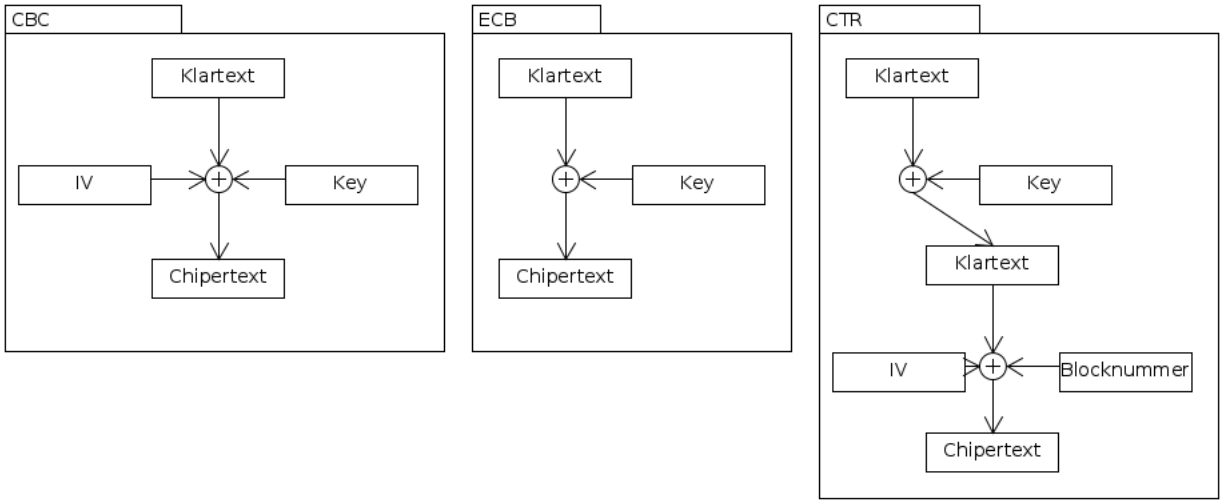
- Vertraulichkeit
- Integrität
- Verfügbarkeit

2 Eigener Verschlüsslungs Algo.

12 Stellige Passwörter werden nach dem Muster abuvcdwxefyz gebildet. Wobei uvwxyz der Passwort Kern &asz6K ist. und ab die ersten beiden Buchstaben der url entsprechen. cd den dritten und vierten Buchstaben welche um ROT+1 verschoben sind. Und schlussendlich ef die letzten beiden Buchstaben welche um ROT-1 verschoben sind. URLs Kürzer als 6 zeichen werden einfach wiederholt.

www.heise.de => heiseh
abuvcdwxefyz
he&ajtszdg6K

3 CBC ECB CTR



4 Anhang

XOR TABLE (HEX)

XOR	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Identities:

$x \oplus 0 = x$ (i.e., $0 \oplus 0 = 0$, $1 \oplus 0 = 1$)
$x \oplus 1 = \neg x$ (i.e., $0 \oplus 1 = 1$, $1 \oplus 1 = 0$)
$x \oplus x = 0$ (i.e., $0 \oplus 0 = 0$, $1 \oplus 1 = 0$)
$x \oplus \neg x = 1$ (i.e., $0 \oplus 1 = 1$)

1.1 IP Tables

1.1.1 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
```

1.1.2 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

1.1.3 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

1.1.4 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

1.1.5 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

1.2 Funktionen & Bedenklichkeiten

Bedenklich	Attacke	Verbesserung
char *gets(char *str);	Buffer Overflow	char *fgets(char *str, int num, FILE *stream);
gets(song)		fgets(song, sizeof(song), stdin)
int sprintf(char *str, const char *fomat, …);	Buffer Overflow	int snprintf(char *s, size_t n, const char * format, …);
int sprintf(command, „get %s.mp3“, song);		int snprintf(command, sozeof(command), „get%s.mp3“, song); command[sizeof(command)-1]='\0‘;
size_t strlen(const char *s);	Buffer Overflow	size_t strnlen(const char *s, size_t maxlen);
len = strlen(str);		len = strnlen_s(str, sizeof str);
char * strcpy (char * destination, const char * source);	Buffer Overflow	size_t strlcpy(char *destination, const char *source, size_t size); ODER char * strncpy(char *destination, const char *source, size_t size);
`strcpy (str2,str1);`		strncpy (str2, str1, sizeof(str2)); len = strlcpy(str2, str1, sizeof(str2));
char *strcat(char *destination, const char *source)	Command Injection	size_t strlcat(char *dst, const char *src, size_t size);
strcat(to, from)		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

1.3 Primzahltabelle mit Hex

Primzahl	2	3	5	7	11	13	17	19	23
Hexwert	2	03	05	07	0b	0d	11	13	17

Primzahl	29	31	37	41	43	47	53	59	61
Hexwert	1d	1f	25	29	2b	2f	35	3b	3d

Primzahl	67	71	73	79	83	89	97
Hexwert	43	47	49	4f	53	59	61

1.4 Sonstige Bedenklichkeiten

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

The ASCII Table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
00	00	NUL	32	20	SP	64	40	@	96	60	‘
01	01	SOH	33	21	!	65	41	A	97	61	a
02	02	STX	34	22	"	66	42	B	98	62	b
03	03	ETX	35	23	#	67	43	C	99	63	c
04	04	EOT	36	24	$	68	44	D	100	64	d
05	05	ENQ	37	25	%	69	45	E	101	65	e
06	06	ACK	38	26	&	70	46	F	102	66	f
07	07	BEL	39	27	’	71	47	G	103	67	g
08	08	BS	40	28	(	72	48	H	104	68	h
09	09	HT	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL