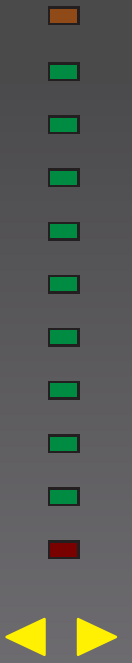
Netzangriffe

Jörg & Mathias & Sebastian

9.8.2002



Inhaltsverzeichnis

1 Scanning

Ermitteln von Angriffspunkten.

1.1 CGI-Portscanner

Benutzt wurde ein CGI-Portscanner. Aus der `/etc/inetd.conf` werden die Ports herausgezogen. Das nützt jedoch nichts, wenn der jeweilige Daemon, der den Dienst unterstützt nicht läuft. Dabei ist die prinzipielle Arbeitsweise von nmap gleich dem des CGI-Portscanner. Letzterer braucht jedoch den laufenden Apache-daemon. Für unser Beispiel wurde in der Datei `/etc/httpd/httpd.conf` folgendes geändert:
`DocumentRoot "/usr/local/httpd/htdocs"`**, da das Orginalunterverzeichnis nicht existent war.**
Zudem wurde in der Zeile
`Options Indexes -FollowSymLinks +Includes MultiViews` **der Eintrag** `+ExecCGI` **ergänzt. Danach brachte ein** `/etc/init.d/apache restart` **den Apache zum laufen und unser CGI-script lief.**

1.2 nmap

Nmap lieferte folgende Ausgabe im Test:

```
root@lotte:~ > nmap -oO hanf.zrz.tu-berlin.de

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on hanf.zrz.TU-Berlin.DE (130.149.5.232):
(The 1526 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     filtered    smtp
79/tcp     open        finger
80/tcp     open        http
98/tcp     open        linuxconf
111/tcp    open        sunrpc
113/tcp    open        auth
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
```

Scanning


```
754/tcp    open         krb_prop
965/tcp    open         unknown
1024/tcp   open         kdm
1026/tcp   open         nterm
5000/tcp   filtered     fics
6000/tcp   open         X11


Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

1.3 scanssh

Das Tool dient laut http://www.openssh.com/usage/ dem Messen der Verbreitung der verschiedenen Secure Shell (ssh) Servern im Internet. Dabei werden zufällige Daten von mit ssh konfigurierten Hosts im Netz gesammelt. Dies geschieht aber nur einmalig. Ziel ist es die Verbreitung der ssh-Tools im Web und auf einer Usenix Konferenz vorzustellen.

```
root@lotte:~/scanssh > ./scanssh 192.168.1.83
192.168.1.83 SSH-1.99-OpenSSH_2.9p2
root@lotte:~/scanssh > ./scanssh avh.dndns.org
Can not parse avh.dndns.org
scanssh: generate failed on avh.dndns.org
scanssh: nothing to scan
root@lotte:~/scanssh > ping avh.dyndns.org
PING avh.dyndns.org (217.230.179.126) from 192.168.1.83 :
                    56(84) bytes of data.
64 bytes from pD9E6B37E.dip.t-dialin.net (217.230.179.126):
                    icmp_seq=1 ttl=243 time=76.415 msec
root@lotte:~/scanssh > ./scanssh 217.230.179.126
```

```
217.230.179.126 SSH-1.5-OpenSSH_2.9.9p2
root@lotte:~/scanssh >
```

Zuerst machten wir einen Programmdurchlauf im lokalen Netz, sodann mit einer externen Adresse. Wie zu ersehen, ist scanssh nicht zu begeistern mit Hostnamen, wodurch man sich mit ping helfen müßte.

2 Sniffing

Abhöhren von Netzwerkverkehr.

2.1 tcpdump

Aufruf:

```
tcpdump [-adeflnNOpqRStvxX] [-c count] [-F file]
        [-i interface] [-m module] [-r file]
        [-s snaplen] [-T type] [-w file]
        [-E algo:secret] [expression]
```

tcpdump zeichnet den Netzwerkverkehr auf. Es ermöglicht eine Vielzahl von Filtern um das Ziel zu spezifizieren bzw. den aufzuzeichnenden Verkehr zu minimieren. Um den gesamten Netzwerkverkehr zu überwachen müsste man zuvor einige spoofing-Methoden einsetzen.

Ein kleines Beispiel:

```
joerg:/home/jobel # tcpdump -Xs 65535 -n
```

Das Ergebnis sieht wie folgt aus:

```
10:38:59.493503 arp who-has 192.168.1.82 tell 192.168.1.94
0x0000   0001 0800 0604 0001 0020 1854 1594 c0a8        ..........T....
0x0010   015e 0000 0000 0000 c0a8 0152                  .^.........R
10:38:59.494077 arp reply 192.168.1.82 is-at 0:0:b4:4e:21:ad
0x0000   0001 0800 0604 0002 0000 b44e 21ad c0a8        ..........N!...
0x0010   0152 0020 1854 1594 c0a8 015e a599 0100        .R...T.....^....
0x0020   0001 0000 0000 0000 076b 6c65 696e             .........klein
10:38:59.494122 192.168.1.94.2173 > 192.168.1.82.23: S 1045564408:1045564408(0)
win 5840 <mss 1460,sackOK,timestamp 497145 0,nop,wscale 0> (DF) [tos 0x10]
0x0000   4510 003c 9e01 4000 4006 18aa c0a8 015e        E..<..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0bf8 0000 0000        ...R.}..>R......
0x0020   a002 16d0 c457 0000 0204 05b4 0402 080a        .....W..........
0x0030   0007 95f9 0000 0000 0103 0300                  ............
```

```
10:38:59.574659 192.168.1.82.23 > 192.168.1.94.2173: P 103:118(15) ack 179
 win 6432 <nop,nop,timestamp 2241307 497154> (DF) [tos 0x10]
0x0000   4510 0043 fae1 4000 4006 bbc2 c0a8 0152        E..C..@.@......R
0x0010   c0a8 015e 0017 087d 4dab ca2e 3e52 0cab        ...^...}M...>R..
0x0020   8018 1920 9164 0000 0101 080a 0022 331b        .....d.......".3.
0x0030   0007 9602 6b6c 6569 6e65 7220 6c6f 6769        ....kleiner.logi
0x0040   6e3a 20                                       n:.


...

10:39:10.090333 192.168.1.82.23 > 192.168.1.94.2173: P 123:133(10) ack 184
 win 6432 <nop,nop,timestamp 2242359 498205> (DF) [tos 0x10]
0x0000   4510 003e fae6 4000 4006 bbc2 c0a8 0152        E..>..@.@......R
0x0010   c0a8 015e 0017 087d 4dab ca42 3e52 0cb0        ...^...}M..B>R..
0x0020   8018 1920 b4bf 0000 0101 080a 0022 3737        ..............".77
0x0030   0007 9a1d 5061 7373 776f 7264 3a20             ....Password:.
```

Sniffing


```
10:39:12.782741 192.168.1.94.2173 > 192.168.1.82.23: P 184:185(1) ack 133
 win 5840 <nop,nop,timestamp 498474 2242359> (DF) [tos 0x10]
0x0000   4510 0035 9e14 4000 4006 189e c0a8 015e        E..5..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0cb0 4dab ca4c        ...R.}..>R..M..L
0x0020   8018 16d0 32ca 0000 0101 080a 0007 9b2a        ....2.........*
0x0030   0022 3737 6b                                   ."77k
10:39:12.996057 192.168.1.94.2173 > 192.168.1.82.23: P 185:186(1) ack 133
 win 5840 <nop,nop,timestamp 498496 2242632> (DF) [tos 0x10]
0x0000   4510 0035 9e15 4000 4006 189d c0a8 015e        E..5..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0cb1 4dab ca4c        ...R.}..>R..M..L
0x0020   8018 16d0 38a2 0000 0101 080a 0007 9b40        ....8.........@
0x0030   0022 3848 64                                   ."8Hd
10:39:13.214720 192.168.1.94.2173 > 192.168.1.82.23: P 186:187(1) ack 133
 win 5840 <nop,nop,timestamp 498518 2242649> (DF) [tos 0x10]
0x0000   4510 0035 9e16 4000 4006 189c c0a8 015e        E..5..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0cb2 4dab ca4c        ...R.}..>R..M..L
0x0020   8018 16d0 377a 0000 0101 080a 0007 9b56        ....7z.........V
0x0030   0022 3859 65                                   ."8Ye
10:39:13.940207 192.168.1.94.2173 > 192.168.1.82.23: P 187:188(1) ack 133
 win 5840 <nop,nop,timestamp 498590 2242671> (DF) [tos 0x10]
```

```
0x0000   4510 0035 9e17 4000 4006 189b c0a8 015e        E..5..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0cb3 4dab ca4c        ...R.}..>R..M..L
0x0020   8018 16d0 2c1b 0000 0101 080a 0007 9b9e        ....,...........
0x0030   0022 386f 70                                   ."8op
10:39:14.153872 192.168.1.94.2173 > 192.168.1.82.23: P 188:189(1) ack 133
 win 5840 <nop,nop,timestamp 498611 2242744> (DF) [tos 0x10]
0x0000   4510 0035 9e18 4000 4006 189a c0a8 015e        E..5..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0cb4 4dab ca4c        ...R.}..>R..M..L
0x0020   8018 16d0 3abc 0000 0101 080a 0007 9bb3        ....:...........
0x0030   0022 38b8 61                                   ."8.a
10:39:14.540192 192.168.1.94.2173 > 192.168.1.82.23: P 189:190(1) ack 133
 win 5840 <nop,nop,timestamp 498650 2242765> (DF) [tos 0x10]
0x0000   4510 0035 9e19 4000 4006 1899 c0a8 015e        E..5..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0cb5 4dab ca4c        ...R.}..>R..M..L
0x0020   8018 16d0 287f 0000 0101 080a 0007 9bda        ....(...........
0x0030   0022 38cd 73                                   ."8.s
10:39:14.711146 192.168.1.94.2173 > 192.168.1.82.23: P 190:191(1) ack 133
 win 5840 <nop,nop,timestamp 498667 2242804> (DF) [tos 0x10]
0x0000   4510 0035 9e1a 4000 4006 1898 c0a8 015e        E..5..@.@......^
0x0010   c0a8 0152 087d 0017 3e52 0cb6 4dab ca4c        ...R.}..>R..M..L
```

```
0x0020   8018 16d0 2846 0000 0101 080a 0007 9beb        ....(F.........
0x0030   0022 38f4 73                                   ."8.s
```

```
10:39:15.636489 192.168.1.82.23 > 192.168.1.94.2173: P 133:600(467) ack 193
 win 6432 <nop,nop,timestamp 2242913 498757> (DF) [tos 0x10]
0x0000   4510 0207 faef 4000 4006 b9f0 c0a8 0152        E.....@.@......R
0x0010   c0a8 015e 0017 087d 4dab ca4c 3e52 0cb9        ...^...}M..L>R..
0x0020   8018 1920 6a61 0000 0101 080a 0022 3961        ....ja........"9a
0x0030   0007 9c45 0d0a 4c61 7374 206c 6f67 696e        ...E..Last.login
0x0040   3a20 4672 6920 4175 6720 2039 2031 303a        :.Fri.Aug..9.10:
0x0050   3335 3a32 3920 3230 3032 2066 726f 6d20        35:29.2002.from.
0x0060   6c6f 6361 6c68 6f73 7420 6f6e 2070 7473        localhost.on.pts
0x0070   2f31 320d 0a4c 696e 7578 206b 6c65 696e        /12..Linux.klein
0x0080   6572 2032 2e34 2e31 392d 7072 6532 2023        er.2.4.19-pre2.#
0x0090   3220 534d 5020 4d69 7420 4de4 7220 3620        2.SMP.Mit.M.r.6.
0x00a0   3031 3a30 373a 3433 2043 4554 2032 3030        01:07:43.CET.200
0x00b0   3220 6936 3836 2075 6e6b 6e6f 776e 2075        2.i686.unknown.u
0x00c0   6e6b 6e6f 776e 2047 4e55 2f4c 696e 7578        nknown.GNU/Linux
0x00d0   0d0a 0d0a 4d6f 7374 206f 6620 7468 6520        ....Most.of.the.
0x00e0   7072 6f67 7261 6d73 2069 6e63 6c75 6465        programs.include
0x00f0   6420 7769 7468 2074 6865 2044 6562 6961        d.with.the.Debia
0x0100   6e20 474e 552f 4c69 6e75 7820 7379 7374        n.GNU/Linux.syst
0x0110   656d 2061 7265 0d0a 6672 6565 6c79 2072        em.are..freely.r
```

```
0x0120   6564 6973 7472 6962 7574 6162 6c65 3b20        edistributable;.
0x0130   7468 6520 6578 6163 7420 6469 7374 7269        the.exact.distri
0x0140   6275 7469 6f6e 2074 6572 6d73 2066 6f72        bution.terms.for
0x0150   2065 6163 6820 7072 6f67 7261 6d0d 0a61        .each.program..a
0x0160   7265 2064 6573 6372 6962 6564 2069 6e20        re.described.in.
0x0170   7468 6520 696e 6469 7669 6475 616c 2066        the.individual.f
0x0180   696c 6573 2069 6e20 2f75 7372 2f73 6861        iles.in./usr/sha
0x0190   7265 2f64 6f63 2f2a 2f63 6f70 7972 6967        re/doc/*/copyrig
0x01a0   6874 0d0a 0d0a 4465 6269 616e 2047 4e55        ht....Debian.GNU
0x01b0   2f4c 696e 7578 2063 6f6d 6573 2077 6974        /Linux.comes.wit
0x01c0   6820 4142 534f 4c55 5445 4c59 204e 4f20        h.ABSOLUTELY.NO.
0x01d0   5741 5252 414e 5459 2c20 746f 2074 6865        WARRANTY,.to.the
0x01e0   2065 7874 656e 740d 0a70 6572 6d69 7474        .extent..permitt
0x01f0   6564 2062 7920 6170 706c 6963 6162 6c65        ed.by.applicable
0x0200   206c 6177 2e0d 0a                              .law...
10:39:15.637222 192.168.1.82.23 > 192.168.1.94.2173: P 600:615(15) ack 193
 win 6432 <nop,nop,timestamp 2242913 498760> (DF) [tos 0x10]
0x0000   4510 0043 faf0 4000 4006 bbb3 c0a8 0152        E..C..@.@......R
0x0010   c0a8 015e 0017 087d 4dab cc1f 3e52 0cb9        ...^...}M...>R..
0x0020   8018 1920 7609 0000 0101 080a 0022 3961        ....v........"9a
```

```
0x0030   0007 9c48 6b64 6540 6b6c 6569 6e65 723a        ...Hkde@kleiner:
0x0040   7e24 20                                        ~$.
```

2.2 dsniff

Aufruf:

```
dsniff [-c] [-d] [-m] [-n] [-i interface] [-s snaplemn]
       [-f services] [-t trigger[,..]] [-r|-w savefile]
       [expression]
```

Beschreibung:

dsniff ist ein Passwort-Sniffer für die meisten gängigen Protokolle. Das interessante ist, dass dsniff automatisch nur die wichtigen Bits filtert und die Ausgabe im Berkley DB Format erfolgt.

Beispielaufruf für eine telnet-Verbindung:

Sniffing


```
jobel@joerg:~> telnet 192.168.1.82
Trying 192.168.1.82...
Connected to 192.168.1.82.
Escape character is '^]'.
Debian GNU/Linux testing/unstable localhost
kleiner login: kde
Password:
Last login: Thu Aug  8 13:05:59 2002 from joerg on pts/20
Linux kleiner 2.4.19-pre2 #2 SMP Mit Mär 6 01:07:43 CET 2002
i686 unknown unknown GNU/Linux

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
kde@kleiner:~$ ls
autosave        favicon.ico.2           nsmail          geheim
Desktop         favicon-t.ico           paddy.midi      Mail
kde@kleiner:~$ exit
logout
Connection closed by foreign host.
jobel@joerg:~>
```

Das Ergebnis des Angriffs ist nicht zu übersehen.
Neben Benutzernamen und Kennwort werden auch alle Befehle ausgegeben.
Siehe unten:

```
joerg:/home/jobel/Documents/dsniff-2.4 # ./dsniff
dsniff: listening on eth0
-----------------
08/08/02 13:06:49 tcp joerg.1137 -> sebastian.23 (telnet)
kde
kdepass
ls
exit
```

Wie gut ist dsniff wirklich?
Die Backspace-Taste wird nicht ausgewertet, wie das nächste Beispiel zeigt.

```
jobel@joerg:~> telnet 192.168.1.82
Trying 192.168.1.82...
Connected to 192.168.1.82.
Escape character is '^]'.
Debian GNU/Linux testing/unstable localhost
kleiner login: kde
Password:
Last login: Thu Aug  8 14:23:06 2002 from joerg on pts/21
Linux kleiner 2.4.19-pre2 #2 SMP Mit Mär 6 01:07:43 CET 2002 i686 unknown
unknown GNU/Linux

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
kde@kleiner:~$ ls
autosave                Mail                        ttt
Desktop                 my                          twinkle.midi
kde@kleiner:~$ exit
logout
Connection closed by foreign host.
jobel@joerg:~>
```

Das Ergebnis sieht dann wie folgt aus:

```
----------------
08/08/02 14:36:25 tcp joerg.1141 -> sebastian.23 (telnet)
kde
peterralfalfkdepassassss
ls
exit
```

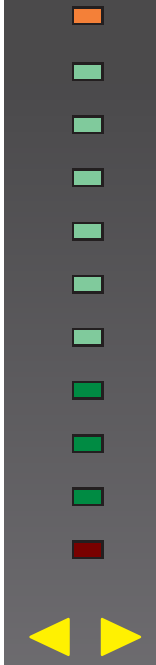
Das Passwort ist immer noch „ kdepass“. Alle eingegebenen Zeichen die mit der Backspace-Taste wieder gelöscht wurden, werden von dsniff mit ausgegeben. Dies ist zwar kein hinreichender Schutz aber der Angreifer muss jetzt schon ein wenig probieren.

3 Spoofing

Angriff durch gefälschte Netzwerkpakete.

3.1 Spoofing bei verbindungslosen Protokollen

Besonders einfach ist Spoofing bei verbindungslosen Protokollen wie z.B. ARP und DNS.



3.1.1 ARP - Spoofing

ARP (Address Resolution Protocol) weist im lokalen Netzwerk die MAC-Adresse der zugehörigen IP-Adresse zu.

Beispielprogramm:

arpspoof

Aufruf:

```
arpspoof [-i interface] [-t target] host
```

Wirkungsweise:

`arpspoof` sendet an das target arp-replies der Form `host` is at `MAC-Adresse des Angreifers`.

Damit werden Verbindungen vom `target` mit dem Angreifer statt mit dem `host` aufgebaut. Läßt man das `target` weg, werden an alle Rechner im lokalen Netz arp-replies geschickt.

Will der Angreifer „nur“ lauschen, so muss er, damit `target` seine Verbindung überhaupt bekommt, die Pakete weiterleiten (forwarden). Unter Linux:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Gegenmaßnahmen:

Feste Einträge im ARP-Cache mit

```
arp -s ipnummer macadresse
```

oder

```
arp -f filename
```

erzeugen.

3.1.2 DNS - Spoofing

DNS (Domain Name System) ordnet die Rechnernamen ihren IP-Nummern zu.

Beispielprogramm:

dnsspoof

Aufruf:

```
dnsspoof [-i interface] [-f hostfile] expression
```

Wirkungsweise:

`dnsspoof` sendet auf Nameserver-Anfragen gefälschte Antworten zurück. Die falschen Namensauflösungen stehen in `hostfile` (beim weglassen, wird auf alle Nameserveranfragen mit der eigenen Adresse geantwortet), `expression` sind `tcpdump`-Auswahl-Expressions (damit kann man sich den Host den man ärgern will, auswählen).

Spoofing

Beim geswitchtem Netz muss kann man es zusammen mit arpspoof benutzen, damit man die Nameserveranfragen der anderen mitbekommt.

Gegenmaßnahmen:

DNS über TCP (falls irgendwo im Angebot, evt. maraDNS)
Nachteil: langsamer, aufwendiger, empfindlich auf DoS-Atacken.

Andere Autentifizierung als DNS (Bsp. echte Zertifikate beim SSL).

3.2 Spoofing bei verbindungsorientierten Protokollen

3.2.1 TCP Verbindungen

In jeder TCP-Verbindung werden TCP-Sequenznummern ausgetauscht und von beiden Partnern bestätigt. Zu Beginn einer Verbindung wird eine möglichst zufällige Sequenznummer zum Start ausgewählt. Hierdurch wird u.a. verhindert, dass ein Angereifer zu einem System eine TCP-Verbindung „simulieren“ kann, da er bei einer falschen IP-Absenderadresse die erste Sequenznummer vom Server nicht erfährt.

Angriffe auf dieser Ebene ist nur möglich, wenn die Sequenznummern ermittelbar sind...

Spoofing

Standardvorgehensweise beim Angriff:

- **Ziel identifizieren**
- **Host, für den er sich ausgeben will, lahmlegen**
- **Adresse des Host vortäuschen**
- **mit Ziel verbinden mit gefälschter Hostadresse**
- **Sequenznummer erraten**

Das Problem ist, das die Antwortpakete des Servers zum Orginalhost geschickt werden, d.h. ich muss die Sequenznummern erraten.

Spoofing

Beispielprogramm:

```
kleiner:~/mendax# ./mendax
Usage: ./mendax [OPTIONS] <source> <target> [<gateway>]

 -p PORT        first port on localhost to occupy
 -s PORT        server port on <source> to swamp
 -l USERNAME    user on <source>
 -r USERNAME    user on <target>
 -c COMMAND     command to execute
 -w PORT        wait for a TCP SYN packet on port PORT
 -d             read data from stdin and send it.
 -t             test whether attack might succeed
 -L TERM        spoof rlogind instead of rshd.
 -S PORT        port from which to sample seq numbers.
```

Wirkungsweise:

Baut TCP-Verbindung zu `source` **auf, gibt sich dabei als** `<target>` **aus schickt Befehl** `COMMAND`**.**

Beispielaufruf:

```
./mendax -p 514 192.168.1.94 192.168.1.83 -l test -r test
flooding source with TCP SYN packets from 143.209.4.3: ....................
sampling sequence numbers...


no detectable difference pattern.
using 30 as prediction difference (0 hits).
spoofing rshd.
resetting TCP target connection: .
resetting source: ....................
```

(Funktionierte bei uns nicht, da Sequenznummern nicht erratbar [zumindest für `mendax`].)

Aktuelles Beispiel (http://www.aerasec.de/security):

System: Raptor Firewall / Symantec Enterprise Firewall, VelociRaptor, Gateway Security

Topic: Vorhersagbare TCP Sequenznummern

Links: Security&Bugrware#5595, Symantec

ID: ae-200208-016

[gekürzt]
Bei Raptor Firewalls ist diese erste Sequenznummer vom Absender- und Zielport abhängig, also nicht vollkommen zufällig gewählt. Speziell wenn es sich bei dem betroffenen Gerät um eine Proxy-Firewall handelt, kann dies ein Problem darstellen. Betroffen hiervon sind die Raptor Firewall 3.5 NT), 3.5.3 (Solaris), Symantec Enterprise Firewall 6.5.2 (Win 2k/NT), 7.0 (Solaris, Win 2k/NT), VelociRaptor 500 - 1300 und die Symantec Gateway Security 5110 - 5300. Symantec bietet einen Patch zur Behebung dieses möglichen Problems.

4 URLs

Scanner:

http://www.monkey.org/~provos/scanssh/
http://www.novia.net/~muesu/PORTSCAN/
http://www.insecure.org/nmap/

Sniffer und Spoofer:

http://www.monkey.org/~dugsong/dsniff/

Sicherheitslöcher:

http://www.aereasec.de/security