Rechnernetze

Security Tools: cops & tiger

Bernd Lehle / Oliver Reutter

Security Tools: cops & tiger

Bernd Lehle / Oliver Reutter

Nachdem in der vorletzten BI.-Ausgabe SATAN als Security Check Tool für komplette Netzwerke vorgestellt wurde, kommen nun zwei "Kollegen" an die Reihe, die einzelne Rechner lokal auf Sicherheitslöcher testen. Insbesondere wollen wir hier auch Wert auf die Einschätzung dieser Tools legen und Einsatzrichtlinien für den Gebrauch in typischen Arbeitsumgebungen vorstellen.

Wie bereits angedeutet, gibt es zwei Klassen von Security Test Tools. Netztester wie SATAN oder ISS sammeln ihre Informationen ausschließlich, indem sie Verbindungen zu anderen Rechner aufbauen und deren Antworten auf die Anforderung bestimmter Dienste auswerten. Lokale Tester wie *cops* oder *tiger* suchen auf dem lokalen Rechner alle verfügbaren Informationen zusammen und werten sie aus.

Die Vor- und Nachteile der beiden Klassen werden hier deutlich: Ein Netztester kann relativ leicht viele Maschinen testen und sowohl Verbindungen, als auch Vertrauen zwischen diesen Maschinen analysieren. Allerdings ist er auf die Antworten der untersuchten Maschinen angewiesen und kann auch dann nur vermuten, warum eine Maschine in einer bestimmten Form antwortet. Der lokale Tester hat Zugriff auf alle Konfigurationsdateien, wie z.B. auch die Paßwort-Datei, was wesentlich detailliertere Aussagen über die Sicherheit eine Systems erlaubt, als dies ein Netztester könnte. Allerdings muß beim Test von vielen Rechnern jeder für sich geprüft werden.

Aus der Sicht eines Angreifers erlaubt es der Netztester die Opfer durch großflächige Scans im Netz herauszusuchen. Hat er dann eine kleine Hintertür im System gefunden (z.B. einen guest Account ohne Paßwort), kann er diesen Account nutzen, um mit dem lokalen Tester von `guest` auf `root` zu kommen. Interessanterweise benutzt die elektronische Unterwelt kaum SATAN, sondern fast nur die Public Domain Version 1.3 von ISS, die klein, kompakt und im Hintergrund lauffähig ist.

Bevor wir zum Handwerklichen der Tools kommen, sei dem Benutzer noch die Einsatzphilosophie ans Herz gelegt, da nicht jeder mit diesen doch recht starken Werkzeugen immer richtig umgeht.

Wenn man ein Tool verwendet und die Aussage *"Keine verwundbaren Stellen gefunden"* geliefert bekommt, ist das allerdings keine absolute Aussage über die Sicherheit des Rechners, sondern bedeutet nur, daß alle in den Datenbanken des Tools gespeicherten Löcher nicht gefunden wurden. Man kann dann bestenfalls sicher sein, daß ein Angreifer, der dasselbe Tool in derselben Konfiguration verwendet, ebenfalls keine Lücken findet. Um ein System nach den Regeln der Kunst abzusichern, empfehlen wir den Einsatz von mehreren Tools beider Klassen, bevor man eine Aussage trifft. Aber auch dann kann man sich nicht zurücklehnen und in Sicherheit wiegen, denn diese Aussagen haben nur eine Halbwertzeit von etwa einem Jahr.

Ebenso sollte man sich überlegen, ob man für den Einsatz die erforderliche Autorisierung hat. SATAN verlangt `root`-Zugriff bevor es startet; man kann damit aber beliebige andere Rechner testen, für die man nicht verantwortlich ist, was die dortigen Betreuer dann oft als Angriff werten. Besondere Vorsicht ist bei Subnetz-Scans angebracht: Es wäre nicht das erstemal, daß ein irrtümlich ausgeführter SATAN-Scan die Alarmglocken bis hoch zum DFN-CERT schrillen läßt und dann auch wie ein Angriff verfolgt wird. Dasselbe gilt natürlich auch für die lokalen Tester. Mit Ausnahme von SecurScan (einem lokalen Sicherheitstester von Silicon Graphics) laufen die meisten ohne `root`-Zugriff, was jedem Benutzer erlaubt, die Sicherheit des eigenen Systems unter die Lupe zu nehmen. Inwiefern sich der Betreuer über diese ungebetene Hilfe freut sei dahingestellt.

Nach Rücksprache kann man seine Systeme auch von unserer Sicherheits-Arbeitsgruppe testen lassen. Wir sichern uns allerdings vorher dahingehend ab, daß alle geprüften Maschinen im Zuständigkeitsbereich des jeweiligen Betreuers liegen und er auch die Verantwortung für eventuelle Folgen übernimmt.

Da sich nun hoffentlich jeder im klaren darüber ist, was ihm in die Hand gegeben wird, können wir zur technischen Seite kommen und die Pakete cops und tiger vorstellen:

- **cops** (Computer Oracle and Password System) wurde von Dan Farmer in seiner Studienzeit an der Purdue University bei Professor Gene Spafford entwikkelt. Dan Farmer ist uns ja von SATAN her bereits bekannt; Gene Spafford hat das Paket **tripwire** entwickelt und auch bereits einige wichtige Bücher zum Thema Sicherheit geschrieben

- **tiger** wurde von Mitarbeitern an der Texas A & M University entwickelt, die es einfach satt hatten, ständig gehackt zu werden und zur Selbsthilfe griffen.

Installation und Konfiguration beider Tools

cops ist eine Sammlung von C- und Perl-Programmen sowie Bourne Shell-Skripten. Maschinennutzer, die System V oder andere nicht 100-prozentigen BSD-Klones einsetzen, sollten vor dem ersten Start das Skript `reconfig` aufrufen, das die Pfade für benötigte Kommandos anpaßt. Es existieren zwei verschiedene Versionen des Programmpakets: Zum einen eine vollständig in Perl geschriebene, zum anderen eine Version, die sich aus C-/Shell-Programmen zusammensetzt. Wir möchten uns hier nur auf die C-/Shell-Version beziehen. Um cops seinen Wünschen anzupassen, muß man das zentrale Shell-Skript cops editieren.

Ein typisches Beispiel einer System V-Maschine sieht dann so aus:

```
#If this is changed to NO, the report that cops creates
#will not be deleted and the results will not be mailed
#to anyone.
```

```
MMAIL=YES

#Foreign language users can change this (thanks to
#Wolfgang Denk!):

LANGUAGE=english
export LANGUAGE

#If this is changed to YES, then the report will only be mailed
#if it detects a difference between the last report & this one.
#Note that this makes no sense unless the mail is set to YES
#as well.

ONLY_DIFF=YES

#Do you want to run suid.chk within cops?

RUN_SUID=NO

#Where is everyone?

ECHO=/bin/echo
TEST=/usr/ucb/test
RM=/bin/rm CAT=/bin/cat
MAIL=/bin/mail
DATE=/bin/date
CHMOD=/bin/chmod
AWK=/bin/nawk
SED=/bin/sed
MV=/bin/mv
MKDIR=/bin/mkdir

#send errors and verbosity to...

BIT_BUCKET=/dev/stdout

# send verbose messages to...

VERBUCKET=/dev/stdout

######################
# Change these lines!#
######################
SECURE=/opt/sonst/cops_104
SECURE_USERS="sneakers@rus.uni-stuttgart.de"
######################
```

(Der Rest wurde gekürzt!)

Danach müssen noch die C-Quelldateien mit `make` oder `make install` übersetzt werden und die Dateien `is_able.lst` und `crc_list` noch den realen Gegebenheiten im eingesetzten System angepaßt werden: Jetzt steht einem ersten Einsatz von cops nichts mehr entgegen!

```
./cops -v -s . -b cops_err
```

Die Datei `./docs/warnings` bietet Hilfe bei der Interpretation der Ergebnisse. Mittels Error Log (hier `cops_err`) kann man Laufzeitproblemen auf die Spur kommen.

tiger bietet im Gegensatz zur C-/Shell-Variante von cops (hier muß das HauptshellSkript editiert werden) ein globales Konfigurationsfile. In diesem kann man den Verlauf des Scans sehr genau den Systemgegebenheiten anpassen.
Hier ein kleiner Auszug aus der Datei `./tigerrc` :

```
# Select checks to perform.Specify "esinglbase;N' (uppercase) for checks
# you don't want performed.
#

Tiger_Check_PASSWD=Y        #Fast
Tiger_Check_GROUP=Y         #Fast
Tiger_Check_ACCOUNTS=Y      #Time varies on # of users
Tiger_Check_RHOSTS=Y        #Time varies on # of users
Tiger_Check_NETRC=Y         #Time varies on # of users
Tiger_Check_ALIASES=Y       #Fast
Tiger_Check_CRON=Y          #Fast
Tiger_Check_ANONFTP=Y       #Fast
Tiger_Check_EXPORTS=Y       #Fast
Tiger_Check_INETD=Y         #Could be faster, not bad though
Tiger_Check_KNOWN=Y         #Fast
Tiger_Check_PERMS=Y         #Could be faster, not bad though
Tiger_Check_SIGNATURES=Y    #Several minutes
Tiger_Check_FILESYSTEM=Y    #Time varies on disk space up to hours
```

Dem letzten Statement im obigen Auszug kommt auf dem Campus eine gewichtige Bedeutung zu. Auf Maschinen, auf denen AFS eingesetzt wird, sollte der FilesystemCheck auf keinen Fall durchgeführt werden, ohne den Zugriff auf AFS abzustellen, da tiger ansonsten die halbe Welt abscannt!

Aufgrund der immensen Einstellungsmöglichkeiten, die dieses Programm bietet, kann hier nur ein kleiner Teil seiner Fähigkeiten beleuchtet werden. Nach dem Auspacken/Anpassen des Konfigurationsfiles sind an für sich keine weiteren Tätigkeiten nötig. Genauso wie in cops kann E-Mail an den System Administrator verschickt werden. Auf kleineren Systemen kann tiger direkt als Cron Job regelmäßig ausgeführt werden, während auf großen bis sehr großen Servern mit dem Skript `tigercron` die Möglichkeit besteht, die einzelnen Scans über mehrere Wochentage zu verteilen.

Für beide Tools existiert die graphische Oberfläche MERLIN, die dem Systemverwalter die Nutzung ähnlich leichtmacht, wie bei der Verwendung von SATAN.

Wo bekommt man die Programme?

Hier zwei wichtige Adressen:

- `ftp://ftp.uni-stuttgart.de/pub/unix/security`
- `ftp://ftp.cert.dfn.de/pub/tools`

Warnung!

Die hier gegebenen Hinweise reichen keinesfalls zum vollständigen Gebrauch beider Tools aus. Jeder, der die Verwendung beider Programme erwägt, sollte gründlichst die beigefügte Dokumentation durchlesen und auch verstehen was er tut.

Für Risiken und Nebenwirkungen lesen Sie bitte die Programm-Dokumentation oder fragen Sie Ihren zuständigen Security Administrator.

Ansprechpartner in sicherheitsrelevanten Fragen

An wen kann ich mich in so einem Fall wenden?
Hier zwei wichtige Adressen und Ansprechpartner:

- sneakers@rus.uni-stuttgart.de
- dfncert-request@cert.dfn.de

Bernd Lehle, NA-5531
E-Mail: Lehle@rus.uni-stuttgart.de

Oliver Reutter, NA-4513
E-Mail: Oliver.Reutter@rus.uni-stuttgart.de