

Werkzeuge für Penetrationstests

Alexander Geschonneck, Berlin*

Mit dem Penetrationstest prüft man ein Sicherheitssystem auf seine Wirksamkeit, indem man – je nach Sicherheitsziel – Dinge tut, die das Sicherheitssystem eigentlich verhindern soll. Dies kann z. B. das Durchdringen einer Firewallkonfiguration, das Lahmlegen eines Servers, das Stehlen von vertraulichen Daten oder die unberechtigte Anmeldung als Systemadministrator sein. Häufig ergeben sich auch unspezifische Fragestellungen wie „Schauen Sie mal, was Sie so alles

mit unseren Systemen aus dem Internet anstellen können.“

Die Ergebnisse eines Penetrationstests stellen eine Momentaufnahme der IT-Sicherheit eines bestimmten Untersuchungsgegenstandes zu einem bestimmten Zeitpunkt dar.

Es ist sinnvoll, bei Planung eines Penetrationstests eine konkrete Fragestellung zu formulieren.

Hier sollte festgelegt werden, was überprüft werden soll. Sind vertrauliche Daten einsehbar? Kann man in das interne Netz eindringen? Welche Informationen sind über die eigene Systemlandschaft im Internet „sicht-

bar“? Können der Internetzugang oder einzelne Komponenten lahmgelegt werden? Wurden die Intrusion Detection Systeme sinnvoll implementiert? Werten die Administratoren die Protokolldateien regelmäßig aus?

Im Folgenden sollen einige Werkzeuge vorgestellt werden, die im Fokus einer konkreten Fragestellung bei einem Penetrationstest sinnvoll sein können. Diese Werkzeuge werden zum Teil auch von Angreifern eingesetzt; es macht also Sinn, dass sie in den gut gefüllten Werkzeugkasten eines jeden Penetrationstesters gehören. Die zum Teil freie Verfügbarkeit im Internet bedeutet, dass jeder po-

* Alexander Geschonneck, ist seit 1998 als Senior Security Consultant bei der HiSolutions AG.

tentielle Angreifer zu diesen Werkzeugen Zugang hat.

Die Auswahl der hier vorgestellten Werkzeuge orientiert sich am Ablauf eines externen Angriffs (dieser soll ja in der Regel bei einem Penetrationstest simuliert werden).

1. Vorhandene Systeme bestimmen.
2. Aktive Dienste und unterstützende Protokolle bestimmen.
3. Schwachstellensuche.

Der Einsatz der hier vorgestellten automatisierten Schwachstellenscanner birgt auch Gefahren.

Die Ergebnisse sollten sorgsam geprüft werden. Häufig melden diese Tools fälschlicherweise eine Schwachstelle, die nicht vorhanden ist. Hier obliegt es der Erfahrung des Testers, diesen Alarm manuell zu verifizieren. Wichtig ist auch, dass sich das Testteam bewusst ist, welche Schwachstellen durch die Tools untersucht werden und welche NICHT. Man darf sich auf keinen Fall nur auf einen

automatisierten Schwachstellenscanner verlassen.

Zusätzlich existieren für verschiedene Teilaspekte eines Penetrationstests separate Programme und Skripte. Der Einsatz von Werkzeugen darf nicht darüber hinwegtäuschen, dass das eigentliche Potential die Kreativität des Testteams ist. Diese darf durch solche Tools nicht eingeschränkt werden. □

```
audit1.hisolutions.com - [audit1] - F-Secure SSH Client
File Edit View Window Help
Quick Connect Profiles
Nmap run completed -- 1 IP address (1 host up) scanned in 9 seconds
[root@audit1 alex]# nmap 192.168.1.1
Starting nmap V. 2.54BETA29 ( www.insecure.org/nmap/ )
Interesting ports on 192.168.1.1:
(The 1535 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open       ftp
23/tcp    open       telnet
25/tcp    open       smtp
53/tcp    open       domain
79/tcp    open       finger
80/tcp    filtered  http
135/tcp   filtered  loc-srv
137/tcp   filtered  netbios-ns
138/tcp   filtered  netbios-dgm
139/tcp   filtered  netbios-ssn
5000/tcp  open       fics
12345/tcp open       NetBus
54320/tcp open       bo2k

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
[root@audit1 alex]# ~
Connected to audit1.hisolutions.com  SSH2 - 3des-cbc - hmac-sha1 - none  79x25  22, 25  00:01
```

Name: nmap (Network Mapper)

Quelle: <http://www.nmap.org/>

Funktion: Multifunktionaler Port- und Protokollscanner, OS-Fingerprinting

Bemerkungen: Dieses Tool gehört zur Grundausstattung eines jeden Penetrationstesters und bietet außer den bei den meisten Port-Scanning-Tools implementierten Funktionen erweiterte TCP- und UDP-Funktionalitäten. Nmap liegt im Quellcode vor.

```
root@audit1:/usr/local/tools
[root@audit1 tools]# xprobe www.ccc.de -v
X probe ver. 0.0.2
Interface: eth0/217.110.47.226
LOG: Target: 213.73.91.29
LOG: Netmask: 255.255.255.255
LOG: probing: 213.73.91.29
LOG: [send]-> UDP to 213.73.91.29:32132
LOG: [98 bytes] sent, waiting for response.
TREE: IP total length field value is OK
TREE: Frag bits are OK
LOG: [send]-> ICMP echo request to 213.73.91.29
LOG: [68 bytes] sent, waiting for response.
TREE: ICMP code != 0
TREE: DF bit in icmp echo response is echoed
TREE: DF bit in icmp unreachable is not echoed
TREE: OpenBSD 2.1-2.3.2.4-2.5!NetBSD 1.5, 1.4.1, 1.4!IBM OS/390
TREE: OpenBSD 2.4-2.5!NetBSD 1.5, 1.4.1, 1.4!IBM OS/390
FINAL: [ OpenBSD 2.4-2.5!NetBSD 1.5, 1.4.1, 1.4 ]
[root@audit1 tools]#
```

Name: Xprobe

Quelle: <http://www.sys-security.com/html/projects/X.html>

Funktion: ICMP Hostfingerprinting

Bemerkungen: Erst in Version 0.0.2 vorliegend, bietet dieses Tool bereits genaue Analysemöglichkeiten anhand verschiedener ICMP-Parameter. Xprobe liegt im Quellcode vor.

Name: whisker

Quelle: <http://www.wiretrip.net/rfp/>

Funktion: Automatisierter
WWW-Serverscanner

Bemerkungen: Flexibles Perl-Skript.
Anpassungen über eigene Analyse-
datenbank möglich. Diverse IDS-Test-
möglichkeiten. Whisker liegt im Quell-
code vor.

```

root@audit1:/usr/local/tools/whisker
[ root@audit1 whisker ]# ./whisker.pl -h www.
-- whisker / v1.4.0 / rain forest puppy / www.wiretrip.net --

= - - - - - =
= Host: www.
= Server: Apache/1.3.19 (Unix)

+ 200 OK: HEAD /_vti_inf.html
+ 200 OK: HEAD /ncl_items.html
+ 200 OK: HEAD /hits.txt
+ 200 OK: HEAD /log.html
+ 200 OK: HEAD /log.txt
+ 200 OK: HEAD /logfile.html
+ 200 OK: HEAD /logfile.txt
+ 200 OK: HEAD /logger.html
+ 200 OK: HEAD /stats.html
+ 200 OK: HEAD /stats.txt
+ 200 OK: HEAD /wwwstats.html
+ 200 OK: HEAD /search/
+ 200 OK: HEAD /shop/
+ 200 OK: HEAD /temp/
+ 200 OK: HEAD /updates/
+ 200 OK: HEAD /passwd.txt
+ 200 OK: HEAD /cgi-bin/passwd.txt
+ 200 OK: HEAD /password.txt
+ 200 OK: HEAD /cgi-bin/password.txt

[ root@audit1 whisker ]#

```

Name: babelweb

Quelle: <http://www.hsc.fr/ressources/ouils/babelweb/>

Funktion: Automatisierter
WWW-Serverscanner

Bemerkungen: In Perl geschriebenes
Analysetool. Testet mit eigenem Spi-
der HTML-Quellcode auf dem Server
nach erweiterten Informationen. Ba-
belweb liegt im Quellcode vor.

```

root@audit1:/usr/local/tools/babelweb-1.0
[ root@audit1 whisker ]# ./whisker.pl -h www.
-- whisker / v1.4.0 / rain forest puppy / www.wiretrip.net --

-----
- babelweb - ultimate web scanner :) - by sa/hsc -
-----

- Start test on www. port 80/tcp

- Identify server
= OS type : [unix]
= Web server type : [apache]
= Web server version : [1.3.19]
= Web version (raw) : [Apache/1.3.19 (Unix)]
= Web server protocol : [HTTP/1.1]
= HEAD give a page : [no]
= Headers of HEAD / : [Date: Sun, 10 Mar 2002 16:17:54 GMT; Server: Apache/1.3.19 (Unix);
|ternates: ("index.xml.de" 1 {type text/xml} {language de}); ("index.xml.en" 1 {type text/xml} {
|language en});|TCN: list|Vary: negotiate,accept-language|Connection: close|Content-Type: text/html]

- Common vuln. CGI scanning

- CGI scanning

- Run several tests on server
= Proxy enable : [no]

```

Name: arirang

Quelle:
[http://www.monkey.org/~pilot/](http://www.monkey.org/~pilot/arirang/index.html)
[arirang/index.html](http://www.monkey.org/~pilot/arirang/index.html)

Funktion: Automatisierter
WWW-Serverscanner

Bemerkungen: Anpassungen über
eigene Analysedatenbank möglich.
Arirang liegt im Quellcode vor.

```

root@audit1:/usr/local/tools/arirang
Don't tell your webserver free from attack    arirang 1.6beta 2001/08/11
                                              2000-2001 by pilot
                                              powered by twwwscan

Starting arirang 1.6beta    http://www.monkey.org/~pilot

www.
Server: Apache/1.3.19 (Unix)

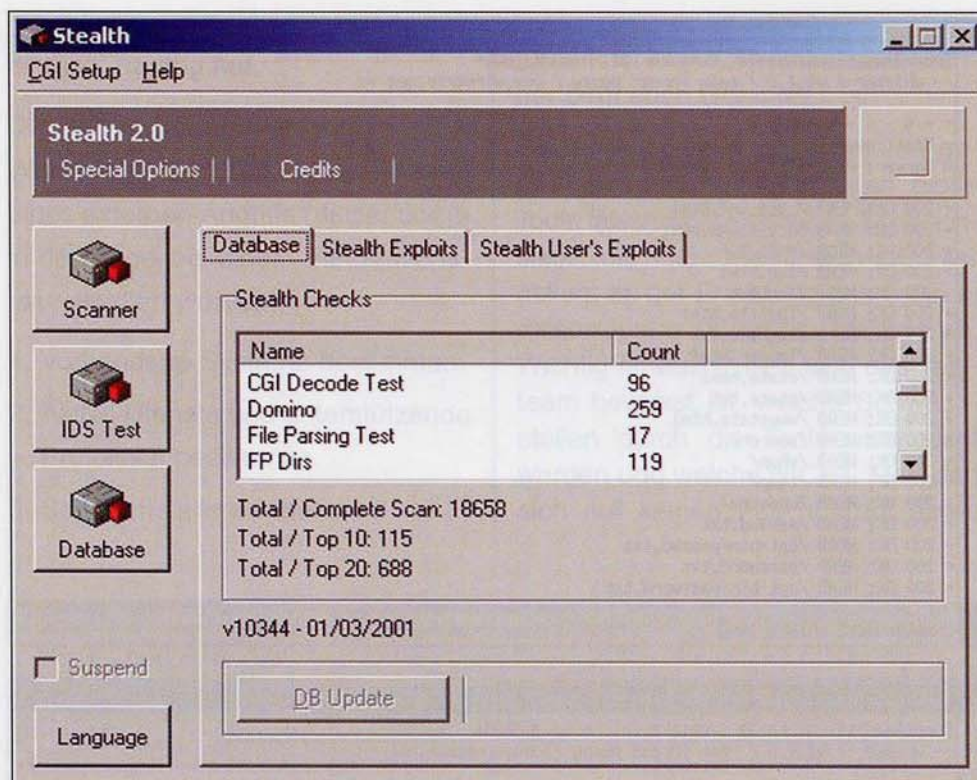
Check Scan all,uxe database

200 OK-Frontpage98 Hole(_vti_inf.html)
FP extensions and the path on the server where the extensions are located.
solution;delete this file

200 OK-Tektronix Webserver(ncl_items.html)

200 OK-Banner Rotation 01(adpassword.txt)

```

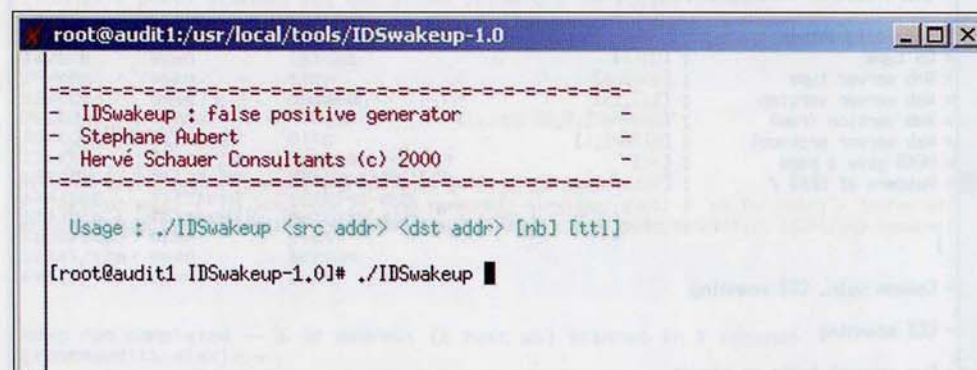


Name: Stealth

Quelle: <http://www.nstalker.com/>

Funktion: Automatisierter WWW-Serverscanner

Bemerkungen: Sehr schneller CGI-Scanner mit multilingualer Oberfläche und umfangreicher Analysedatenbank mit diversen IDS-Testmöglichkeiten. In der Testversion keine automatische Schwachstellenaktualisierung möglich. Nur Windowsversionen verfügbar.

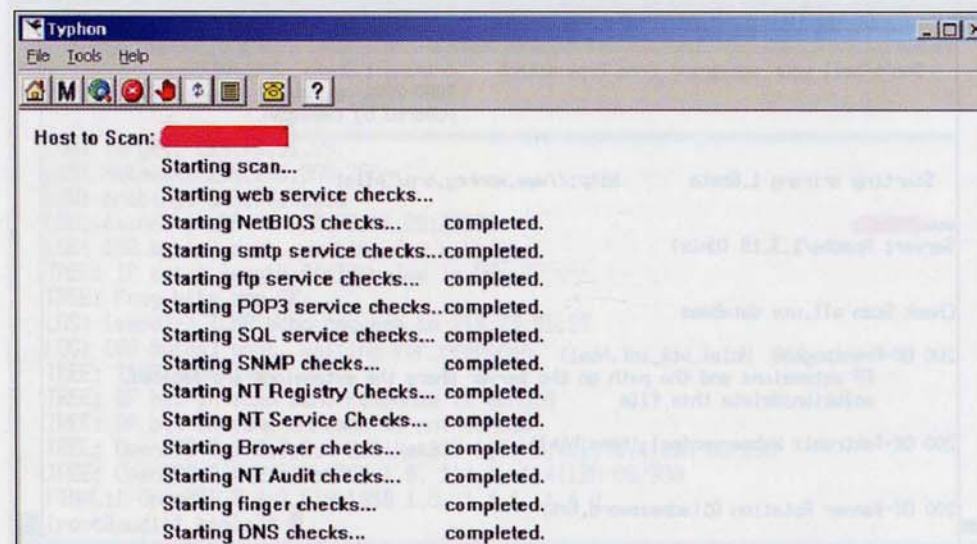


Name: IDSwakeup

Quelle: <http://www.hsc.fr/ressources/outils/>

Funktion: IDS-Tester

Bemerkungen: Simuliert Angriffe im Netz zum Test von Intrusion Detection Systemen. IDSwakeup ist im Quellcode verfügbar.



Name: Typhon

Quelle: <http://www.nextgenss.com/>

Funktion: Automatisierter Schwachstellenscanner

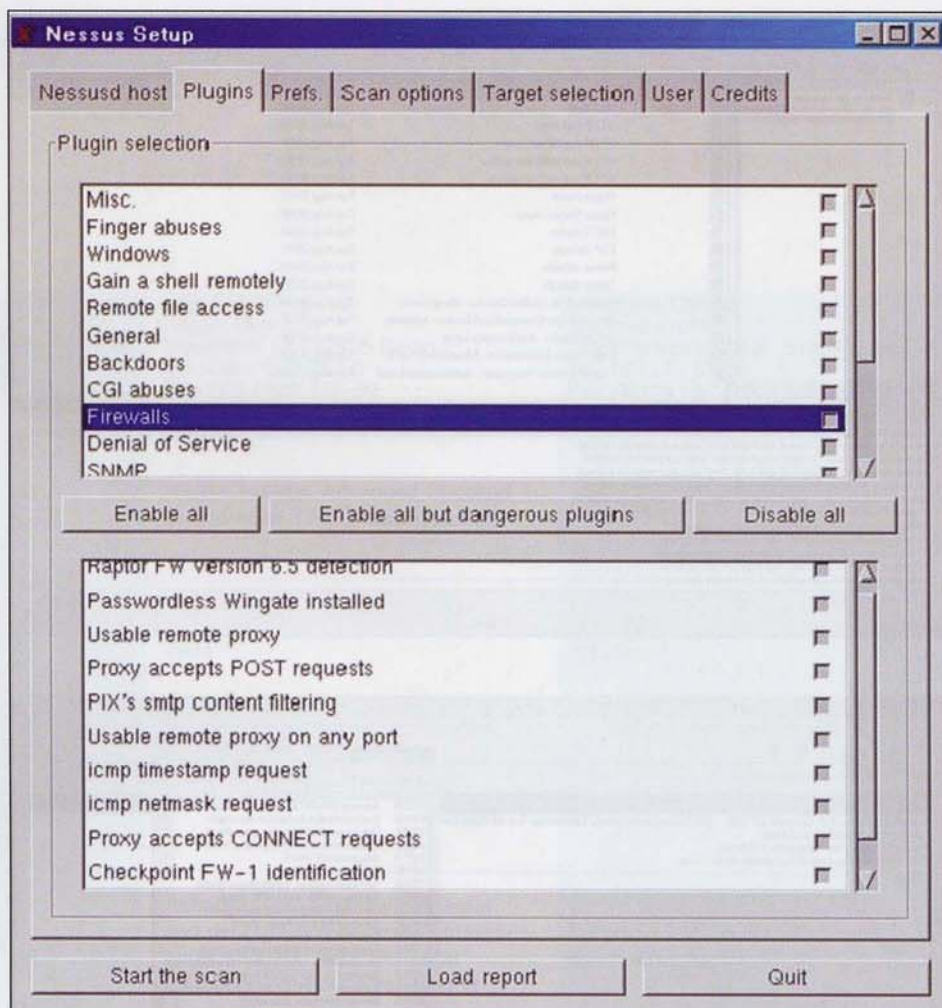
Bemerkungen: Nachfolger des Cerberus Internet Scanners (CIS) mit sehr schneller Analyse gängiger Schwachstellen. 30 Tage Testversion verfügbar. Nur in Windows verfügbar.

Name: Nessus

Quelle: <http://www.nessus.org/>

Funktion: Automatisierter
Schwachstellenscanner

Bemerkungen: Sehr flexibler Schwachstellenscanner. Einbindung von nmap als Portscanner. Automatische Aktualisierung der Schwachstellendatenbank möglich. Tests durch eigene Skriptsprache erweiterbar. Gehört zur Grundausstattung eines jeden Testers. Nessus liegt im Quellcode vor.

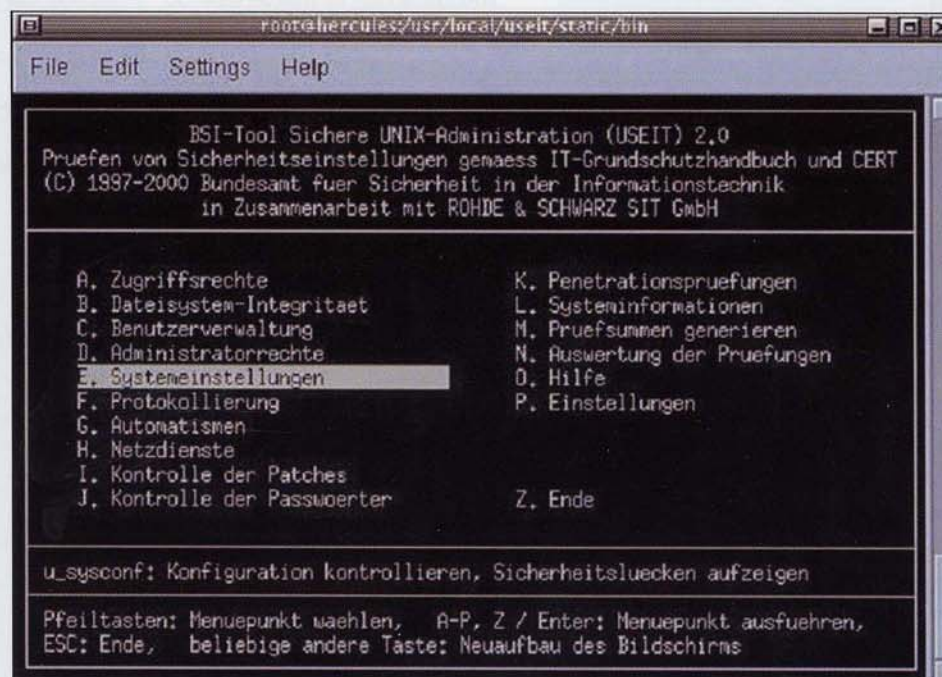
**Name: USEIT**

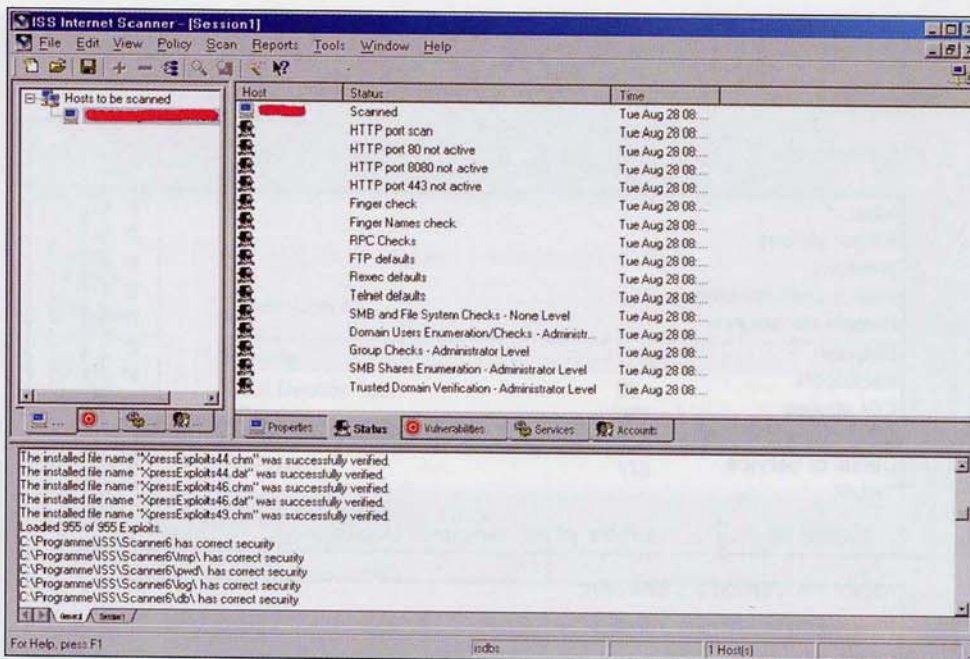
Quelle:
<http://www.bsi.de/aufgaben/projekte/useitool/useit.htm>

Funktion: Automatisierter Schwachstellenscanner

Bemerkungen: Im Auftrag des BSI entwickeltes Sicherheitsanalysewerkzeug für Unix.

Auf jeweilige Versionen und Distributionen angepasst. Testversionen verfügbar.





Name: Internet Security Scanner

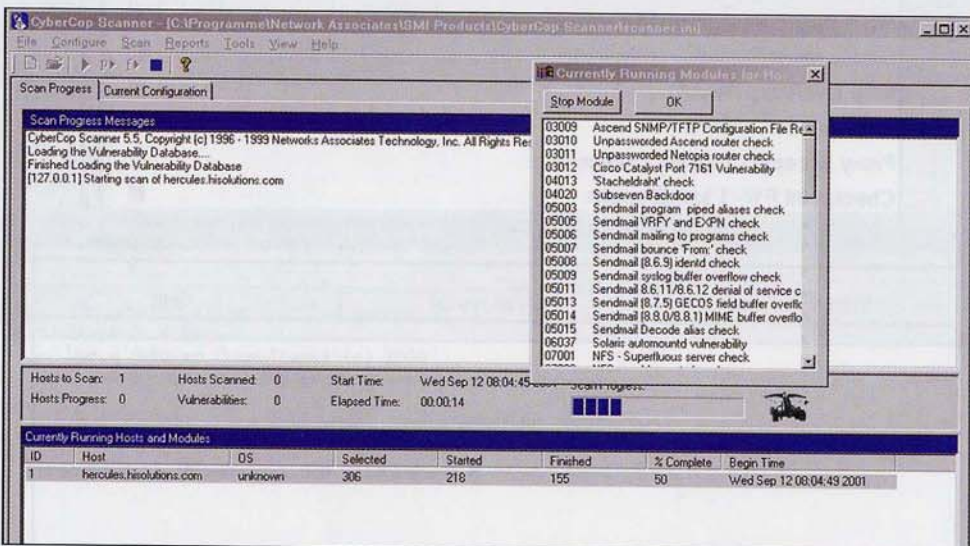
Quelle: <http://www.iss.net/>

Funktion: Automatisierter Schwachstellenscanner

Bemerkungen: Mächtiges Analysewerkzeug mit umfangreicher Schwachstellendatenbank.

Speicherung der Scansession für spätere Analyse in Datenbank möglich.

Automatische Aktualisierung der Schwachstellendatenbank möglich. Unix-Versionen werden nicht mehr gepflegt. Testversionen verfügbar.



Name: CyberCop

Quelle: <http://www.nai.com>

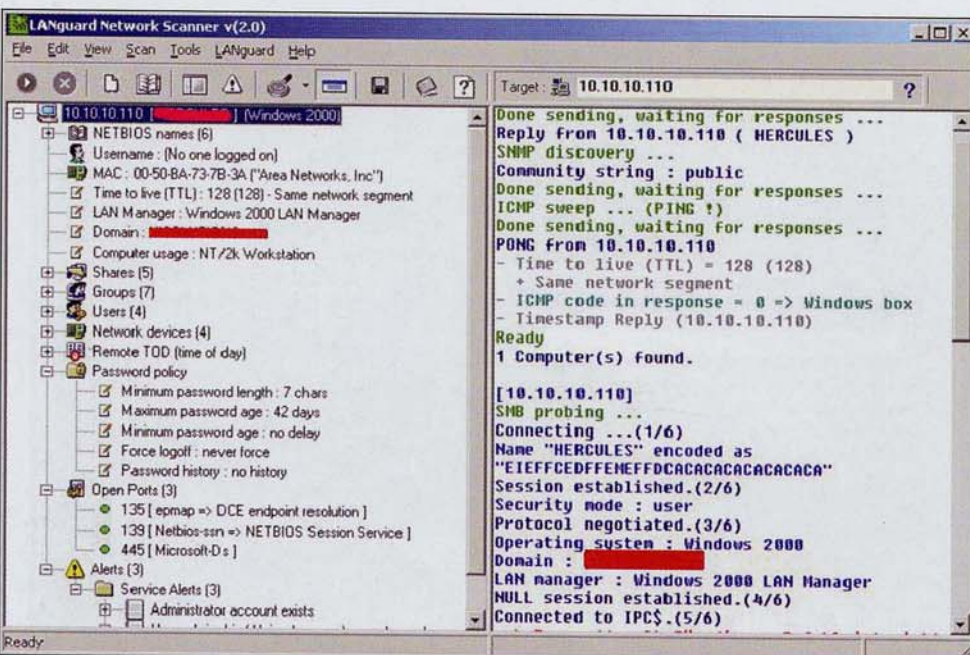
Funktion: Automatisierter Schwachstellenscanner

Bemerkungen: Mächtiges Analysewerkzeug mit umfangreicher Schwachstellendatenbank.

Speicherung der Scansession für spätere Analyse in Datenbank möglich.

Automatische Aktualisierung der Schwachstellendatenbank möglich.

Unix-Versionen werden nicht mehr gepflegt. Testversionen verfügbar.



Name: LANguard Network Scanner

Quelle: <http://www.gfi.com>

Funktion: Automatisierter Schwachstellenscanner

Bemerkungen: Sehr schnelle Analyse gängiger Schwachstellen mit einigen Einschränkungen.

Schwachstellendatenbank anpassbar.

Nur in Windows verfügbar.