

OpenVAS-Client Benutzerhandbuch



\$Date: 2006-05-19 09:58:15 \$

Renaud Deraison <renaud@nessus.org>

Jan-Oliver Wagner <jan@intevation.de>

Deutsche Übersetzung:

Frank Koormann <frank@intevation.de> und

Jan-Oliver Wagner <jan@intevaion.de>

Inhaltsverzeichnis

1	Überblick	4
2	Installation	6
2.1	Aus dem Quelltext	7
2.2	Debian GNU/Linux	7
2.2.1	Debian “Woody” 3.0	7
2.2.2	Debian “Sarge” 3.1	7
2.2.3	Debian “Etch”	7
2.3	RedHat Linux	7
2.3.1	Fedora Core 4 und 5	7
2.4	SUSE Linux	8
2.4.1	SUSE Linux 9.2	8
2.5	MS Windows	8
2.5.1	Windows XP SP2	8
2.6	Migration des Nessus GTK Klient von Version 2.2	8
3	Der erste Report	9
3.1	Starten von OpenVAS-Client	9
3.2	Der Nessus Scan Assistent	9
4	Grundlagen von OpenVAS-Client	9
4.1	Das Hauptfenster	9
4.1.1	Aufgaben	10
4.1.2	Bereiche	11
4.1.3	Reporte	12
4.2	Authentifizierung	13
4.3	Scan Optionen	15
4.3.1	Allgemein	15
4.3.2	Plugins	16
4.3.3	Zugangsdaten	19
4.3.4	Plugin Voreinstellungen	21
4.3.5	Zielauswahl	23
4.4	Reports	24
4.4.1	Report Seite von OpenVAS-Client	24
4.4.2	Report Formate	24
5	Besondere Funktionen	25
5.1	OpenVAS-Client Voreinstellungen	25
5.1.1	Benutzerschnittstelle	25
5.1.2	Verbindung mit Nessus Server	25
5.1.3	Plugin Zwischenspeicher	26
5.1.4	Report	27
5.1.5	Externe Verweise in HTML/PDF	27
5.2	Benutzer-definierte Zugriffsregeln	27
5.2.1	Allgemein	27
5.2.2	Syntax	27
5.2.3	Regeln in OpenVAS-Client verwalten	28
5.2.4	Beispiel für eine Benutzer-definierte Zugriffsregel	28

5.2.5	Beispiel für eine Administrator-definierte Zugriffsregel	28
5.3	Lokale Sicherheitstests: Theorie	28
5.3.1	Was sind "lokale" Sicherheitstests?	28
5.3.2	Welche Betriebssysteme werden derzeit unterstützt?	29
5.3.3	Nach welchen Solaris Patches wird geprüft?	29
5.3.4	Ist es geplant weitere Systeme zu unterstützen?	29
5.3.5	Was sind lokale Sicherheitsprüfungen NICHT?	29
5.3.6	Warum benötigt man lokale Sicherheitstests, ist Nessus nicht gut genug, die Sicherheitsprobleme selbst zu entdecken? . . .	30
5.3.7	Kann Nessus individuell compilierte und installierte Pakete er- kennen?	30
5.3.8	Warum um lokale Sicherheitstests kümmern, wenn es gar keine lokalen Benutzer gibt?	30
5.3.9	Wie kann das Einschalten der lokalen Sicherheitstests die Scan- Erfahrung verbessern?	31
5.3.10	Verlangsamt diese Funktionalität den Scan?	32
5.3.11	Was sind die Nachteile dieser Funktionalität?	32
5.4	Lokale Sicherheitstests: Wie man sie anschaltet	32
5.4.1	Das Prinzip	32
5.4.2	Einen öffentlichen SSH-Schlüssel erzeugen	33
5.4.3	Ein Benutzerkonto einrichten und einen SSH Schlüssel herstellen	33
5.4.4	Nessus aufsetzen	34
5.4.5	Danksagung	34
6	Optimierung und Feinabstimmung	34
6.1	Wissensbasis	34
6.1.1	Einführung	36
6.1.2	Verwendung der WB Speicherung	36
6.1.3	Reduzierung der getesteten Hosts in der Wissensbasis	36
6.1.4	Wiederverwendung der Wissensbasis für Mehrfachtests	37
6.1.5	Bedeutung von aktuell	38

Über dieses Dokument

Dieses Dokument beschreibt die Verwendung des OpenVAS oder Nessus Server über die GTK+ GUI "OpenVAS-Client". Die Beschreibung deckt unter Umständen nicht die gesamte Funktionalität von OpenVAS oder Nessus Server ab, noch erhebt sie Anspruch auf Vollständigkeit.

Bitte beachten Sie, dass es eine Namensänderung von NessusClient nach OpenVAS-Client etwa Anfang Juli 2007 gab. NessusClient 1.x wird seitdem als OpenVAS-Client weiterentwickelt während der vollständig neu entwickelte NessusClient 3.x auf QT anstatt GTK+ basiert und auch keine Freie Software mehr ist.

Aufgrund dieser Namensänderung können in diesem Handbuch noch entsprechende Inkonsistenzen vorliegen.

1 Überblick

Das "Nessus"-Projekt bietet einen leistungsfähigen, aktuellen und leicht benutzbaren Security-Scanner als Freie Software unter der GNU General Public Licence (GPL). Mit Nessus können Sie Computer-Netzwerke im Fernzugriff prüfen und festzustellen, ob die "bösen Jungs" (auch 'Cracker') eindringen oder es anderweitig missbrauchen können.

Intelligentes Scannen Im Gegensatz zu vielen anderen Security-Scannern trifft Nessus keine Voraussetzungen. Z.B sind Scans einzelner Dienste nicht an bestimmte Ports gebunden: Wenn Sie etwa einen WWW-Server an Port 1234 betreiben, so wird Nessus ihn entdecken und auf Sicherheitslücken überprüfen. Ebenso werden Sicherheitslücken nicht einfach anhand von Versionsnummern installierter Dienste geschätzt, sondern es wird versucht, vermeintliche Sicherheitslücken tatsächlich auszunutzen. Damit kann geprüft werden ob Fixes eine Lücke auch tatsächlich geschlossen haben.

Modulare Architektur Die Klient/Server Architektur von Nessus erlaubt eine flexible Installation von Scanner (der Server) und Benutzeroberfläche (der Klient) in vielfältigen Konfigurationen. Dadurch reduzieren sich Installations- und Betreuungsaufwand: Ein Server kann von vielen Klienten benutzt werden.

Weitere Funktionen des Nessus Security Scanner:

Plugin Architektur Jeder Sicherheitstest ist als externes Plugin implementiert. Dadurch können leicht eigene Tests hinzugefügt werden, ohne direkt in den Code des Nessus Servers (`nessusd`) einzugreifen. Eine vollständige Liste der verfügbaren Plugins findet sich unter <http://cgi.nessus.org/plugins>.

CVE kompatibel Jedes Plugin enthält einen Bezug zum CVE-Verzeichnis (Common Vulnerabilities and Exposures) für Administratoren für weitere Informationen zu veröffentlichten Sicherheitslücken. Plugins bieten außerdem Verweise zu **CERT**, **Bugtraq** und Meldungen von Herstellern.

NASL Der Nessus Security-Scanner enthält NASL (Nessus Attack Scripting Language), eine Programmiersprache speziell für die einfache und schnelle Entwicklung von Sicherheitstests. Diese Tests können auch in C implementiert werden.

Aktuelle Datenbank von Sicherheitslücken/-tests Das Nessus-Projekt konzentriert sich auf die Entwicklung von Tests für aktuelle Sicherheitslücken. Die Datenbank der Sicherheitstests wird täglich aktualisiert, die neuesten Tests sind unter <http://www.nessus.org/scripts.php> sowie dem FTP Server und seinen Spiegelserversn verfügbar.

Unbegrenzte Anzahl simultaner Tests Abhängig von der Leistungsfähigkeit des Rechners auf dem der Nessus Server läuft können viele Host-Rechner gleichzeitig überprüft werden.

Flexible Dienste-Erkennung Nessus verlässt sich nicht darauf, dass der zu überprüfende Host-Rechner den von der IANA vorgegebenen Portnummern entsprechend konfiguriert ist. Ein FTP-Server wird auch an einem "Nicht-Standard-Port" erkannt (z.B. an Port 31337), ebenso ein WWW-Server an Port 8080.

Mehrfache Dienste Nessus beendet Scans nicht, sobald ein zu prüfender Dienst an einem Port gefunden wurde: Laufen auf dem Host-Rechner z.B. zwei (oder noch mehr) WWW-Server an Port 80 und 8080, so wird Nessus alle einzeln testen.

Kooperation von Tests Die von Nessus durchgeführten Tests nutzen von anderen Tests bereits vorliegende Informationen: Bietet ein FTP-Server beispielsweise keinen anonymen Login (durch einen Test ermittelt), so werden keine weiteren Tests ausgeführt, die ausschließlich anonyme Logins betreffende Sicherheitslücken überprüfen.

Umfangreiche Berichte Nessus listet nicht nur auf, welche Lücken entdeckt wurden, sondern bietet, soweit verfügbar, auch Hinweise, wie die Lücken geschlossen werden können. Zur Bewertung der Dringlichkeit von Gegenmaßnahmen wird jede Sicherheitslücke einem Risiko-Level zugeordnet (von niedrig bis sehr hoch).

Export von Berichten Der Nessus-Klient kann Berichte als XML, HTML, ASCII, L^AT_EX, PDF und leicht zu parsende Dateien exportieren.

Volle SSL-Unterstützung Nessus kann SSL-bezogene Dienste https, smtps, imaps, u.a. testen. Darüber hinaus kann Nessus mit einem Zertifikat versehen werden und so in PKI-basierten Umgebungen eingesetzt werden.

Intelligente Plugins (optional) Nessus erkennt, welche Tests auf einen Host-Rechner angewandt werden sollten. Dies verhindert z.B. dass Sendmail-Sicherheitslücken auf einem Mail-Server mit Postfix untersucht werden. Diese Option nennt sich "Test optimieren".

Nicht-destruktive Scans (optional) Der oben beschriebene Ansatz, Sicherheitslücken tatsächlich auszutesten, kann dazu führen, dass Dienste abstürzen. Soll dies vermieden werden so können "Sichere Prüfungen" durchgeführt werden. Nessus verlässt sich dann auf Kennungen der Dienste anstatt durch Testen von Lücken zu untersuchen, ob tatsächlich ein Sicherheitsproblem vorliegt.

Unabhängige Entwickler Die Nessus-Entwickler sind nicht abhängig von anderen Software-Herstellern. Es gibt keinen Grund, eine Sicherheitslücke in Programm XYZ zu verschweigen.

Erreichbarkeit der Entwickler Sie vermissen eine Funktion in Nessus? Nehmen Sie einfach Kontakt mit den Entwicklern auf, unter <http://www.nessus.org/contact/> oder auf der Nessus Mailing-Liste. Wir antworten und sind offen für sinnvolle Erweiterungen.

Transparentes Bug-Tracking-System. Sie haben einen Bug gefunden? Melden Sie ihn bitte unter <http://bugs.nessus.org>.

Umfangreiche Tests Es gibt mehr als 7000 Nessus Tests (und es werden ständig mehr!), die in 23 verschiedenen Gruppen unterteilt sind:

- Hintertüren (Backdoors)
- CGI Lücken
- CISCO
- Denial of Service (DoS)
- finger Lücken
- Firewalls
- FTP
- Erlangung einer remote-Shell
- Erlangung von Root-Rechten (remote)
- Allgemein
- Verschiedenes
- Netware
- NIS
- Port Scanner
- Remote Dateizugriff
- RPC
- Einstellungen
- SMTP Probleme
- SNMP
- Ungetestet
- Nutzlose Dienste
- Windows
- Windows: Benutzer-Verwaltung

2 Installation

Bitte beachten Sie, dass die getestete Veröffentlichung des Nessus Entwickler-Teams der Quelltext von Nessus ist. Binär-Pakete sind sehr wahrscheinlich für die meisten Standard-Betriebssysteme erhältlich, aber nicht notwendigerweise durch das Nessus Entwickler-Team getestet.

Für Betriebssysteme die hier nicht aufgelistet sind können trotzdem durchaus gut gepflegte Pakete vorliegen.

2.1 Aus dem Quelltext

Der übliche Weg einer ganz grundlegenden Installation beginnt mit den Quellen als CVS Snapshot oder einem tar-Archiv (tar-ball) des Quelltextes. Mit der Kommando-Sequenz “./configure ; make ; make install” läßt sich das Modul konfigurieren, kompilieren und installieren. Wie üblich enthält das Paket Dateien mit Detail-Informationen zur Konfiguration und Installation. Lesen Sie diese bitte zunächst! Erst danach sollten Sie mit der Installation fortfahren.

2.2 Debian GNU/Linux

Debian unterstützt Nessus-Pakete aktiv. Bitte lesen Sie die Debian Dokumentation zur Installation von Standard-Paketen.

2.2.1 Debian “Woody” 3.0

Die offizielle Version von Nessus für Woody ist 1.0.10. Diese ist vollkommen veraltet und es sollte weder der Nessus Server noch Klient in dieser Version verwendet werden!

OpenVAS-Client hat anspruchslöse Abhängigkeiten zu anderen Paketen während der Kompilation und zur Laufzeit. Daher könnte es leicht möglich sein, sogenannte “Backports” für Woody von neueren Debian Quell-Paketen von OpenVAS-Client herzustellen.

2.2.2 Debian “Sarge” 3.1

Die offizielle Version von Nessus für Sarge ist 2.2.3. OpenVAS-Client ist kompatibel zu diesem Nessus Server, aber es ist empfehlenswert Nessus Server wegen Fehlerkorrekturen auf neuere Versionen zu aktualisieren.

OpenVAS-Client hat anspruchslöse Abhängigkeiten zu anderen Paketen während der Kompilation und zur Laufzeit. Daher könnte es leicht möglich sein, sogenannte “Backports” für Sarge von neueren Debian Quell-Paketen von OpenVAS-Client herzustellen.

2.2.3 Debian “Etch”

Aktuell ist Etch die Testing-Version von Debian, also im fluß. Mindestens Nessus Version 2.2.7 ist enthalten. Es steht zu erwarten, dass OpenVAS-Client Pakete verfügbar werden, sobald OpenVAS-Client veröffentlicht wird.

2.3 RedHat Linux

Im Download-Bereich von www.nessus.org finden Sie evtl. RPM Pakete für RedHat und Fedora welche von Tenable Network Security bereitgestellt werden.

2.3.1 Fedora Core 4 und 5

Diese Distribution führt aktiv eigene Nessus Pakete im Extras-Bereich. Wenn Sie eine neuere Version als die aktuell angebotene benötige, so finden Sie auch hier Backports. Bei der Suche nach entsprechenden RPM-Pakete können Ihnen rpmfind.net oder ähnliche Dienste helfen.

Es existiert zum aktuellen Zeitpunkt noch kein bei Fedora gepflegtes Paket für OpenVAS-Client.

OpenVAS-Client hat anspruchslöse Abhängigkeiten zu anderen Paketen während der Kompilation und zur Laufzeit. Daher könnte es leicht möglich sein, sogenannte “Backports” von OpenVAS-Client herzustellen oder sogar die Pakete direkt zu installieren.

2.4 SUSE Linux

Im Download-Bereich von www.nessus.org finden Sie evtl. RPM Pakete für SUSE welche von Tenable Network Security bereitgestellt werden.

2.4.1 SUSE Linux 9.2

Die aktuellste Version der Nessus Serie 2.0 ist Teil der Distribution. Es wird dringend empfohlen auf die aktuellste Version der 2.2 Serie zu aktualisieren. Wenn Sie eine neuere Version benötigen, so finden Sie auch hier Backports. Bei der Suche nach entsprechenden RPM-Pakete können Ihnen rpmfind.net oder ähnliche Dienste helfen.

Es existiert zum aktuellen Zeitpunkt noch kein bei SUSE gepflegtes Paket für OpenVAS-Client.

OpenVAS-Client hat anspruchslöse Abhängigkeiten zu anderen Paketen während der Kompilation und zur Laufzeit. Daher könnte es leicht möglich sein, sogenannte “Backports” von OpenVAS-Client herzustellen oder sogar die Pakete direkt zu installieren.

2.5 MS Windows

2.5.1 Windows XP SP2

Für Windows ist ein reguläres Setup-Paket für OpenVAS-Client verfügbar. Es sollte die Bezeichnung *OpenVAS-ClientN.N.N-setup-LL.exe* haben. Rufen Sie den Installer direkt auf oder benutzen Sie den Windows Software-Manager. mit letzterem können Sie Nessus auch leicht und vollständig wieder entfernen. N.N.N entspricht der Version von Nessus, z.B. 1.0.0. LL steht für die Sprache wobei “en” die englische Version ist, “de” die deutsche und “sv” die schwedische.

2.6 Migration des Nessus GTK Klient von Version 2.2

OpenVAS-Client nutzt anderes Konzept, “Session”-Daten zu speichern als der Nessus GTK Client der in Nessus 2.2 enthalten war. Tatsächlich wurde das gesamte Konzept von Aufgabe (task), Bereich (scope) und Report (report) grundlegend erneuert.

Der Hauptunterschied: Nessus GTK Klienten bis hin zu Version 2.2 beachten zur Konfiguration eine einzelne Datei “.nessusrc” im Heimat-Verzeichnis des Benutzers. Neuere Versionen nutzen eine Verzeichnis-Struktur mit dem Hauptverzeichnis “.nessus” im Heimat-Verzeichnis des Benutzers.

Dennoch treten keine Konflikte auf, wenn Sie eine neue Nessus-Version starten: Eine vorhandene Datei “.nessusrc” wird als Vorlage für die globalen Einstellungen (siehe unten) genutzt. Die Konfiguration Ihres Nessus Klient 2.2 wird also die Voreinstellung für OpenVAS-Client.

Selbst wenn Sie anschließend wieder einen Nessus GTK Klient 2.2 starten, kann dieser weiterhin die erhalten gebliebenen Einstellungen in der Datei “.nessusrc” nutzen.

3 Der erste Report

Dieser Abschnitt dient als kurze Einführung für neue Anwender in die Arbeit mit Nessus. Die notwendigen Schritte zur Durchführung eines ersten Sicherheits-Scans werden dargestellt. Es wird empfohlen, nach einem ersten erfolgreichen Scan die weitere Abschnitte zur Nutzung von OpenVAS-Client durchzuarbeiten.

3.1 Starten von OpenVAS-Client

Auf jedem UNIX-artigem System können Sie einfach “OpenVAS-Client” in einer Shell eingeben. Selbstverständlich sollte X Windows laufen.

Für Debian GNU/Linux, SUSE Linux, RedHat/Fedora und MS Windows werden zusätzlich Menü-Einträge in der Benutzeroberfläche erzeugt, wenn Sie eines der oben dargestellten Installations-Pakete benutzen.

3.2 Der Nessus Scan Assistent

Der Nessus Scan Assistent kann Sie als Anfänger bei einem ersten Scan unterstützen. Zweck des Assistenten ist es, alle für einen Scan notwendigen Parameter logisch und selbsterklärend zusammen zu fassen.

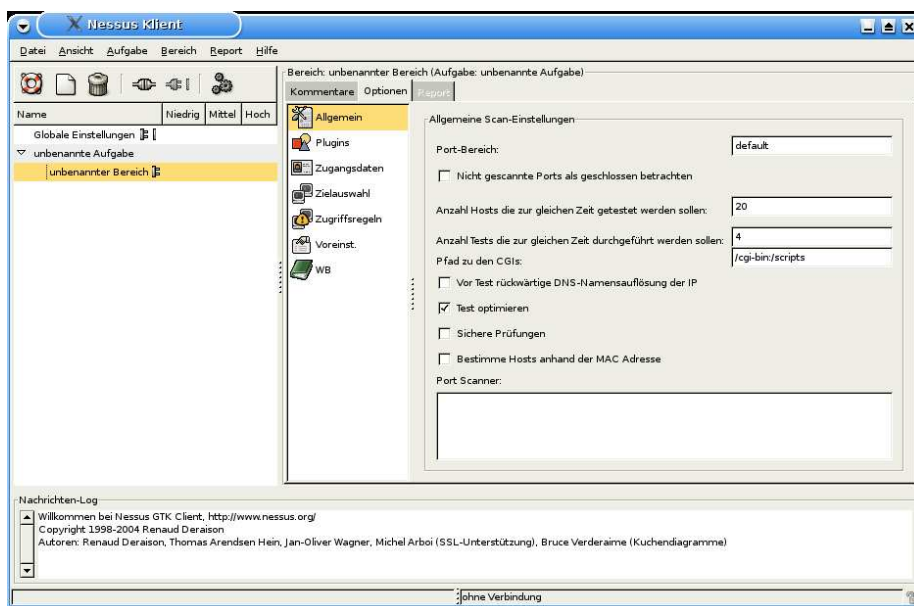
Dennoch sollten Sie beachten, dass ein Scan durchgeführt mit Unterstützung des Assistenten auf den explizit gespeicherten globalen Einstellungen basiert. Wenn Sie in den globalen Einstellungen bereits einige gefährliche Optionen aktiviert haben, so werden diese auf den konkreten Scan vererbt!

4 Grundlagen von OpenVAS-Client

Dieser Abschnitt stellt alle grundlegenden Elemente von OpenVAS-Client vor und beschreibt die typische Anwendung des Klient. Spätere Abschnitte beschreiben spezielle Funktionen des Klienten und richten sich an fortgeschrittene Anwender.

4.1 Das Hauptfenster

Das Hauptfenster von OpenVAS-Client teilt sich in zwei wichtige Bereiche: Links bietet eine Baumstruktur eine Übersicht lokal gespeicherter Aufgaben, Bereiche und Reports. Im rechten Bereich finden sich Karteikarten mit Kommentaren, Einstellungen und Reports. Hier lassen sich Scans einstellen und kommentieren und Ergebnisse sichten. Beachten Sie, dass die Karte der Einstellungen selbst wieder eine Sammlung von Karten enthält.



Nach dem ersten Start von OpenVAS-Client enthält die Baumstruktur nur einen Eintrag: Globale Einstellungen. Dies sind die Voreinstellungen für OpenVAS-Client. Diese berücksichtigen keine bestimmte Auswahl an Plugins, dafür ist eine Verbindung zu einem Nessus Server notwendig. Sie müssen eine Verbindung zu einem Server aufbauen und dann die Einstellungen für eine Auswahl von Plugins angeben. Speichern Sie dann Ihre bevorzugten Voreinstellungen über den Menüpunkt "Globale Einstellungen speichern" des Menüs "Datei".

4.1.1 Aufgaben

Der Sinn von Aufgaben ist es, alle Aktionen eines Oberthemas zusammen zu fassen. Oberthemen sind z.B. "Test aller Rechner am Standort ABC" oder "Kunde XYZ GmbH".

Einer Aufgabe kann ein Kommentar zugeordnet werden, der das Oberthema detaillierter beschreibt. Hier können auch weitere wichtige Informationen oder Hinweise angegeben werden: Etwa wann die nächste Serie von Scans durchgeführt werden soll oder auf welcher Vertragsgrundlage die Scans durchgeführt werden.

Eine Aufgabe hat keine Optionen oder Reporte. Sie stellt nur eine Sammlung von Bereichen dar.

Mögliche Aktionen bezüglich Aufgaben:

Neu Erzeugt eine neue Aufgabe mit dem Titel "unbenannte Aufgabe".

Umbenennen Der Titel einer Aufgabe kann entweder direkt in der Baumstruktur durch Anklicken des Titels oder durch Auswahl des entsprechenden Menü-Eintrags geändert werden.

Löschen Löschen einer Aufgabe bedeutet das Löschen aller zugehörigen Bereiche! Daher erfolgt eine Sicherheitsabfrage vor der eigentlichen Löschung.

4.1.2 Bereiche

Ein Bereich kann als Teil-Aufgabe angesehen werden: Er bezeichnet einen bestimmten Sicherheits-Scan. Der Titel sollte aussagekräftig das Ziel des Scans beschreiben, z.B.: “Behutsamer Scan des produktiven Web-Servers”, “Aggressiver Scan des Web-Server Test-Systems” oder “Alle Sun Workstations”.

Zu jedem Bereich kann ein Kommentar gespeichert werden, der die Bedeutung des Scans detaillierter beschreiben und sonstige hilfreiche Informationen enthalten kann.

Ein Bereich umfasst auch einen vollständigen Satz an Einstellungen für den Sicherheits-Scan. Wenn ein neuer Bereich erstellt wird, werden zunächst die “Allgemeinen Einstellungen” kopiert. Die einzelnen Optionen werden weiter unten genauer beschrieben. Für jeden Bereich kann eine Verbindung zu einem Nessus Server aufgebaut werden. Anschließend kann die Auswahl von Plugins als Teil der Einstellungen durchgeführt werden. Ein Symbol rechts neben dem Titel zeigt die aktive Verbindung zu einem Nessus Server an. Sinn der Zuordnung eines Servers zu einem Bereich ist, dass jeder Server unterschiedliche Plugins anbieten kann. Mit dem Aufbau der Verbindung bezieht OpenVAS-Client eine Liste der verfügbaren Plugins.

Ein Bereich kann außerdem eine Sammlung von Reports umfassen. Jedes Mal, wenn ein Scan erfolgreich durchgeführt wurde, wird ein neuer Report angelegt. zusätzlich können Reports aus Dateien oder von einem Nessus Server geladen werden.

Bitte beachten Sie, dass Änderungen an einem Bereich werden immer und nur genau dann gespeichert wenn ein Scan gestartet wird. Machen Sie Änderungen beispielsweise an der Plugin-Auswahl und schliessen dann OpenVAS-Client ohne einen Scan durchzuführen, so werden die Änderungen verworfen.

Mögliche Aktionen bezüglich Bereiche:

Ausführen Die Konfiguration für den Bereich wird gespeichert und ein Sicherheits-Scan wird mit den aktuellen Einstellungen durch den verbundenen Nessus Server ausgeführt.

Neu Ein neuer Bereich mit dem Titel “unbenannter Bereich” wird als Teil der aktuell ausgewählten Aufgabe angelegt. Als Voreinstellung werden die allgemeinen Einstellungen kopiert. Beachten Sie hier, dass nur explizit gespeicherte Global Einstellungen als Voreinstellungen verwendet werden. Änderungen die nur im OpenVAS-Client vorliegen, haben keine Auswirkung für einen neuen Bereich.

Umbenennen Der Titel eines Bereichs kann entweder direkt in der Baumstruktur durch Anklicken des Titels oder durch Auswahl des entsprechenden Menü-Eintrags geändert werden.

Löschen Löschen eines Bereichs bedeutet das Löschen aller zugehörigen Reports! Daher erfolgt eine Sicherheitsabfrage vor der eigentlichen Löschung.

Verschieben zu Aufgabe Ein Bereich mit allen zugeordneten Reports kann von einer Aufgabe zu einer anderen verschoben werden. Ein Untermenü enthält die Liste aller Aufgaben, hieraus kann das Ziel ausgewählt werden.

Öffnen Ein Bereich kann geladen und der aktuellen Aufgabe zugeordnet werden. Mit diesem Schritt werden jedoch nur die Einstellungen für den Bereich geladen! Die Reports liegen in eigenen Dateien vor, die separat importiert werden können. Die Aktionen Öffnen und Speichern (siehe unten) erlauben es, Einstellungen für einen bestimmten Bereich an andere weiterzugeben oder selbst Sicherungskopie anzulegen.

Speichern als Der aktuellen Bereich wird in eine Datei gespeichert (in der Struktur von nessusrc). Es werden nur die Einstellungen des Bereichs gespeichert, nicht die zugehörigen Reports.

4.1.3 Reporte

Ein Report ist das Ergebnis eines Sicherheits-Scans. Er enthält die Ergebnisse der ausgeführten Plugins zusammen mit Informationen zu Sub-Netz, Zielrechner, Port und Schwere der festgestellten Lücke.

Durch OpenVAS-Client verwaltet kann zusätzlich ein Kommentar angegeben werden. Soweit verfügbar werden auch die zum Scan gehörenden Einstellungen gespeichert. Diese zusätzlichen Informationen sind nicht Teil der Nessus Report Dateien und gehen daher verloren, wenn ein Report exportiert wird! Umgekehrt enthalten auch importierte Reports keine Kommentare oder die zum Report gehörenden Einstellungen.

Mögliche Aktionen bezüglich Reports:

Löschen Löscht den aktuellen Report und zugehörige Kommentare und Einstellungen. Es erfolgt eine Sicherheits-Abfrage vor der Löschung.

Import Importiert einen Report aus einer Datei. Der Standard für Austauschformate ist NBE (Dateiendung “.nbe”). Ältere Versionen von Nessus nutzten das Format NSR (“.nsr”). Es wird empfohlen, ausschließlich das NBE-Format zu benutzen. Über den Dateiauswahl-Dialog kann der gewünschte Report ausgewählt werden. Konnte der Report nicht importiert werden, so erfolgt eine Fehlermeldung. Sonst wird der Report dem aktuellen Bereich hinzugefügt. Beachten Sie, dass weder Kommentare noch Einstellungen aus einer NBE-Datei importiert werden.

Export Der aktuell ausgewählte Report kann exportiert werden. Entweder in einem Nessus-Austauschformat (NBE oder das veraltete NSR) oder in einem Format zur Weiterverarbeitung bzw. Darstellung der Ergebnisse (XML, eine veraltete XML Variante, HTML, HTML mit Diagrammen, \LaTeX , ASCII Text und PDF). Es wird empfohlen, das NBE-Format zu nutzen, wenn die Daten mit anderen ausgetauscht werden sollen und das PDF für einfache Berichte, die keiner weiteren Überarbeitung bedürfen. Die anderen Formate eignen sich zur Weiterverarbeitung der Ergebnisse bzw. zur Übernahme in eigene Dokumente.

Drucken Das Druck-Kommando erzeugt einen PDF-Report und einen auf dem System installierten PDF-Betrachter zu starten. Mittels dessen Druckfunktionen kann der Bericht dann gedruckt werden. Sollte trotz eines installierten Betrachters kein Programm gestartet werden, prüfen Sie bitte, ob sich die Installation in den aktuellen Suchpfaden befindet.

4.2 Authentifizierung

OpenVAS-Client benötigt eine Verbindung zu einem Nessus Server für die Abfrage der Liste verfügbarer Plugins und zum tatsächlichen Lauf eines Sicherheits-Scans.

OpenVAS-Client kann viele Verbindungen zu verschiedenen Servern gleichzeitig halten. Jedem Bereich ist eine eigene Verbindung zu einem Server zugeordnet. Zusätzlich kann für die allgemeinen Einstellungen ein Server verbunden sein, um die Voreinstellung für Plugin-Auswahl und -Parameter zu bestimmen. Beachten Sie hier, dass nur explizit gespeicherte Globale Einstellungen als Voreinstellung für neue Bereich verwendet werden.

The screenshot shows a Windows-style dialog box titled "Verbinden mit Nessus Server". It contains three main sections. The first section, "Nessus Server", has a "Host-Name" field with the text "galene" and a "Port" field with the text "1241", followed by a "Vorgabe" button. The second section, "Authentifizierung", has a "Login" field with the text "thomas" and a "Passwort" field with masked text "xxxx". The third section, "SSL Optionen", contains two checkboxes: "Verwende SSL Verschlüsselung" (checked) and "Authentifizierung durch Zertifikat" (unchecked). Below these are three file selection fields: "Vertrauenswürdige CA:" with the path "/some/path/CA/cacert.pem", "Datei Benutzer-Zertifikat:", and "Datei Benutzer-Schlüssel:", each with an "Auswählen ..." button. At the bottom of the dialog are two buttons: "Abbrechen" (with a red X icon) and "OK" (with a green checkmark icon).

Der Status der Verbindung zum Server wird durch ein Symbol neben dem Titel eines Bereichs oder den allgemeinen Einstellungen in der Baumstruktur dargestellt. Nur für Bereiche kann eine Verbindung zu Nessus Servern aufgebaut werden, nicht für Aufgaben oder Reports.

Weitere Informationen zum Verbindungsstatus zeigt die Status-Zeile am Fuß des Hauptfensters: Hier werden die tatsächlichen Verbindungsdaten angezeigt, z.B. "Verbindung: username@host.test.example". Ganz unten rechts ist ein Icon zu sehen welches anzeigt ob die bestehende Verbindung verschlüsselt ist oder nicht, ganz so wie es die meisten Web-Browser auch anzeigen.

Der "Verbinden"-Dialog bietet verschiedene Einstellungsmöglichkeiten für den Aufbau einer Verbindung zu einem Nessus Server:

Host-Name Der Domain-Name oder die IP-Adresse des Rechners, auf dem der Nessus Server läuft.

Port Der Port an dem der Nessus Server auf Verbindungen wartet. Ältere Nessus Server benutzten Port 3001, der offizielle Port ist nun 1241. Mit dem "Vorgabe"-Knopf können Sie diesen immer wieder einfach einstellen.

Login Ihr Nutzernamen auf dem ausgewählten Host-Rechner. Um einen bestimmten Nessus Server benutzen zu können benötigen Sie ein Nutzer-Konto für diesen Server. Der Administrator des Servers kann ein Konto einrichten.

Passwort Das Passwort für Ihr Nutzer-Konto auf einem Nessus Server. Verwenden Sie hier kein Passwort, das Sie auch für andere Nutzer-Konten verwenden. Insbesondere dann nicht, wenn Sie auch unverschlüsselte Verbindungen zu einem Nessus Server aufbauen.

SSL Optionen

Verwende SSL Verschlüsselung: Es gibt zwei grundsätzliche Methoden der Authentifizierung: via Login/Passwort Kombination oder mittels Zertifikaten (mit oder ohne Passwort). In jedem Fall sollten Sie SSL Verschlüsselung aktivieren, damit Passwörter nicht im Klartext durch das Netzwerk transferiert werden. Die Verschlüsselung ist jedoch für Fälle abschaltbar, in denen kein SSL verfügbar ist, aber trotzdem mit Nessus gearbeitet werden soll.

Vertrauenswürdige CA: Das Zertifikat benennt eine Certificate Authority (CA) der Sie vertrauen. Mit diesem Zertifikat überprüfen Sie, ob Sie sich mit einem vertrauenswürdigen Nessus Server verbinden. Die Überprüfung wird für die Paranoia Level 2 und 3 durchgeführt, nicht jedoch für den Level 1. Der Paranoia Level kann manuell in der Konfigurationsdatei .nessusrc eingestellt werden. Wenn Sie zum ersten Mal eine Verbindung zu einem Nessus Server aufbauen werden Sie außerdem ausdrücklich nach dem gewünschten Paranoia Level gefragt.

Der vorgegebene Pfad für die vertrauenswürdige CA der Dateiname der lokalen Nessus Server Installation. Wenn Sie also Nessus Klient und Server auf dem gleichen Rechner laufen lassen oder beide das gleiche Dateisystem nutzen, so können Sie diese Vorgabe einfach benutzen.

Wenn Sie Nessus remote benutzen, so benötigen Sie eine Kopie des CA Zertifikats, die Sie an beliebiger Stelle in Ihrem Heimat-Verzeichnis ablegen können.

Authentifizierung durch Zertifikate: Wenn Sie diese Methode benutzen, benötigen Sie ein für Sie erstelltes Schlüssel/Zertifikat Paar. Dieses erstellt üblicherweise der Administrator des Nessus Servers mit den entsprechenden Skripten. Sie erhalten zwei Dateien: Benutzer-Zertifikat und Benutzer-Schlüssel. Der Schlüssel kann mit einem Passwort versehen sein, dieses geben Sie dann bitte in das entsprechende Textfeld ein.

4.3 Scan Optionen

Dieser Abschnitt erläutert die wichtigsten Konfigurations-Optionen für einen Sicherheits-Scan (auf der Karte "Optionen"). Speziellere Optionen (z.B. Zugriffsregel, von der Wissensbasis gelöste Scans) werden später im Kapitel über spezielle Funktionen erläutert

4.3.1 Allgemein

Diese Karte enthält alle allgemeinen Scan-Einstellungen, vergleichen Sie dazu auch den Screenshot des Hauptfensters oben.

Port-Bereich Ports die durch Nessus Server gescannt werden sollen. Es können entweder ein Port-Bereich angegeben werden, wie z.B. "1-8000" oder auch komplexe Mengen wie "21,23,25,1024-2048,6000". "-1" bedeutet keine Port-Scans durchzuführen, "default" bedeutet die der Nessus Dienste Datei angegebenen Ports zu scannen.

Nicht gescannte Ports als geschlossen betrachten Um Zeit zu sparen können Sie dem Nessus Server vorgeben, TCP-Ports die nicht gescannt wurden als geschlossen zu betrachten. Die Überprüfung des Zielsystems ist damit unvollständig aber die Scan-Zeit reduziert sich. Außerdem schickt der Nessus Server keine Pakete an Ports die Sie nicht vorgegeben haben. Wenn diese Option nicht aktiviert ist, betrachtet der Nessus Server Ports deren Status unbekannt ist als offen.

Anzahl Hosts die zur gleichen Zeit getestet werden sollen Geben Sie hier die maximale Anzahl an Hosts an, die durch den Server gleichzeitig gescannt werden sollen. Beachten Sie, dass der Nessus Server für einen Scan max_hosts x max_tests Prozesse startet!

Anzahl Tests die zur gleichen Zeit durchgeführt werden sollen Geben Sie hier die maximale Anzahl an Tests an, die durch den Server gleichzeitig gegen jeden Host gestartet werden. Beachten Sie, dass der Nessus Server für einen Scan max_hosts x max_tests Prozesse startet!

Pfad zu den CGIs Nessus bietet die Möglichkeit, mehrere Pfade ("/cgi-bin", "/cgis", "/home-cgis" usw.) nach CGIs zu durchsuchen. Geben Sie die zu durchsuchenden Pfade durch Doppelpunkte getrennt an: Z.B. "/cgi-bin:/cgi-aws:/~deraison/cgi".

Vor Test rückwärtige DNS-Namensauflösung der IP Ist diese Option eingeschaltet, so wird Nessus Server eine DNS Rückwärts-Suche über die IP-Adresse durchführen (DNS reverse lookup) bevor die Tests durchgeführt werden. Dies kann die Durchführung der Tests insgesamt verlangsamen.

Test optimieren Sicherheits-Tests bitten unter Umständen den Nessus Server darum, nur dann gestartet zu werden, wenn bestimmte Informationen die durch andere Plugins gesammelt wurden, sich in der Wissen-Basis befinden oder nur wenn ein angegebener Port offen ist. Diese Option kann die Tests beschleunigen, kann aber dazu führen, dass Nessus Server einige Lücken übersieht. Wer paranoid ist, schaltet diese Option aus.

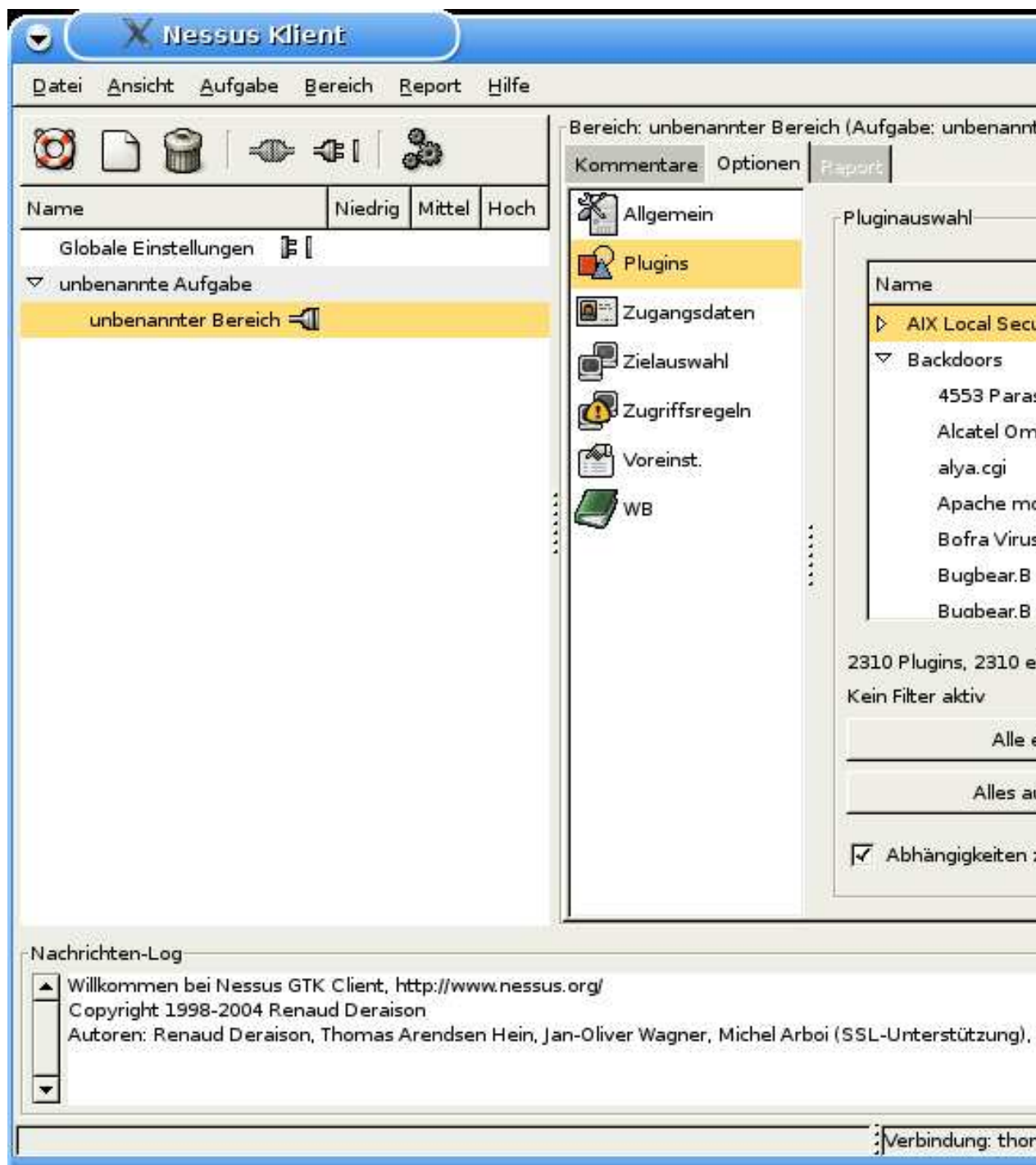
Sichere Prüfungen Einige Sicherheits-Tests können dem Ziel-Rechner Schaden zufügen, indem Dienste zeitweise oder bis zu einem Neustart dort nicht mehr zur Verfügung stehen. Wird diese Option eingeschaltet, so wird Nessus Server sich auf gesendete Banner verlassen und keine echten Tests durchführen. Die Ergebnisse sind dann entsprechend weniger zuverlässig, aber wenigstens wird dadurch kein möglicher Ausfall für Benutzer entstehen. Unter Sicherheitsaspekten ist es angeraten, diese Option auszuschalten. Aus der Sicht eines Administrators sollte sie angeschaltet bleiben.

Bestimme Hosts anhand der MAC Adresse Wird diese Option eingeschaltet, so werden die Ziele im lokalen Netzwerk anhand ihrer Ethernet MAC Adresse anstatt der IP-Adresse bestimmt. Dies ist insbesondere dann sinnvoll wenn Nessus in einem DHCP-Netzwerk verwendet wird. Sind Sie sich unsicher, dann schalten Sie die Option ab.

Port Scanner Die Liste verfügbarer Port-Scanner. Post-Scanner stellen eine besondere Kategorie von Plugins dar und werden daher getrennt von den anderen Plugins dargestellt. Die Liste ist nur gefüllt, wenn eine Verbindung zu einem Nessus Server hergestellt wurde. Sie können einen oder mehrere Scanner auswählen und durch Anklicken mehr Details zu einem ausgewählten Scanner abfragen.

4.3.2 Plugins

Plugins werden auf dem Nessus Server gespeichert. Deshalb benötigen Sie eine Verbindung zu einem Nessus Server, um einen Auswahl zutreffen. Ansonsten wird eine leere Liste angezeigt.



Plugins werden unterschieden in verschiedene Familien, die als Ganzes durch entsprechende Check-Boxen aktiviert oder deaktiviert werden können. Auch kann eine Familie aufgeklappt werden, so dass die zugehörigen Plugins aufgelistet werden und einzeln über die Check-Boxen aktiviert oder deaktiviert werden können um die Aus-

wahl zu verfeinern.

Die Spalte “Warnung” enthält ein Warnungs-Symbol für einige Plugins. Dieses Warnungs-Symbol bedeutet, dass das Plugin Dienste auf dem Zielsystem abschalten oder das gesamte System abschalten könnte. Sie sollten sehr vorsichtig sein, wenn Sie diese Option einschalten, da es nötig werden könnte einige Zielsystem manuell neu zu starten.

Unterhalb der Plugin-Liste wird die Gesamtzahl der vom Server geladenen Plugins angezeigt. Desweiteren auch die Gesamtzahl der ausgewählten Plugins und die Anzahl der durch einen Filter dargestellten Plugins.

Die Folgenden Aktionen sind möglich:

Alle einschalten Schaltet alle Plugin-Kategorien ein.

Alle ausschalten Schaltet alle Plugin-Kategorien aus.

Alles aufklappen Klappt die Plugin-Liste soweit auf, dass sämtliche Plugins angezeigt werden.

Alles zusammenklappen Es werden nur noch die Plugin Familien angezeigt.

Abhängigkeiten zur Laufzeit berücksichtigen Wird diese Option aktiviert, dann wird Nessus Server alle Plugins einschalten die abhängig sind von den bereits selektierten.

Stille Abhängigkeiten Wird diese Option aktiviert, dann wird Nessus Server keinen Bericht für solche Plugins senden, die nicht explizit eingeschaltet waren.

Filter Der Filter-Dialog erlaubt die Auswahl von Plugins anhand bestimmter Muster. **Beachten Sie**, dass Sie die vorherige Plugin-Auswahl komplett löschen sobald Sie einen Filter aktivieren.

Plugin Info-Dialog Wenn Sie auf einen Plugin-Titel einen Doppel-Click ausführen öffnet sich ein Dialog, der weitere Informationen zum Plugin bietet. Die angezeigten Informationen sind im Plugin abgelegt.

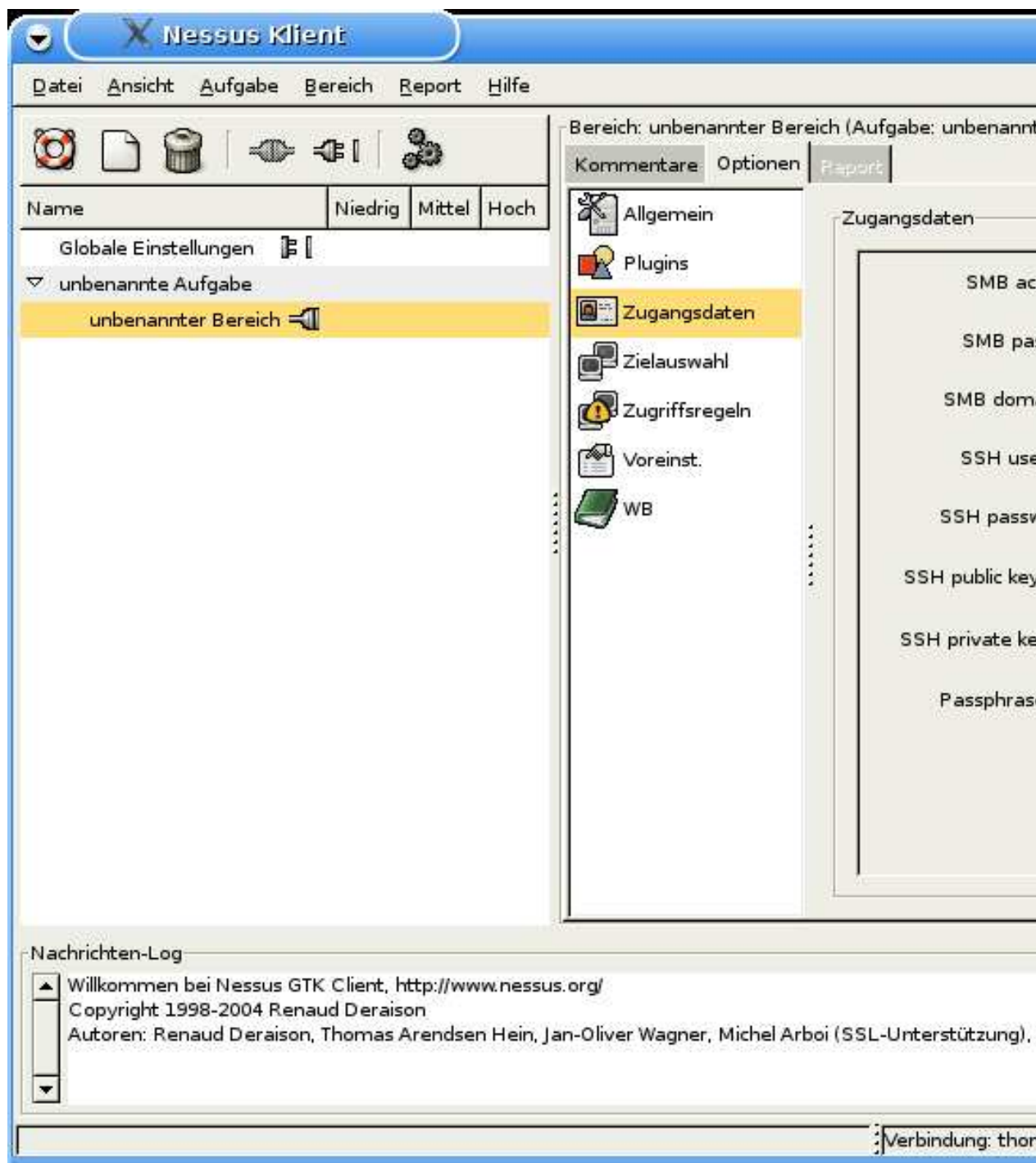
Folgende Aktionen sind im Info-Dialog möglich:

Setze Plugin Timeout Sie können eine Timeout für das Plugin angeben.

Zeige Abhängigkeiten Ein weiterer Info-Dialog listet die Abhängigkeiten des Plugins. Hier wird auch der Status (eingeschaltet/ausgeschaltet) der Plugins angegeben von denen das aktuelle Plugin abhängt.

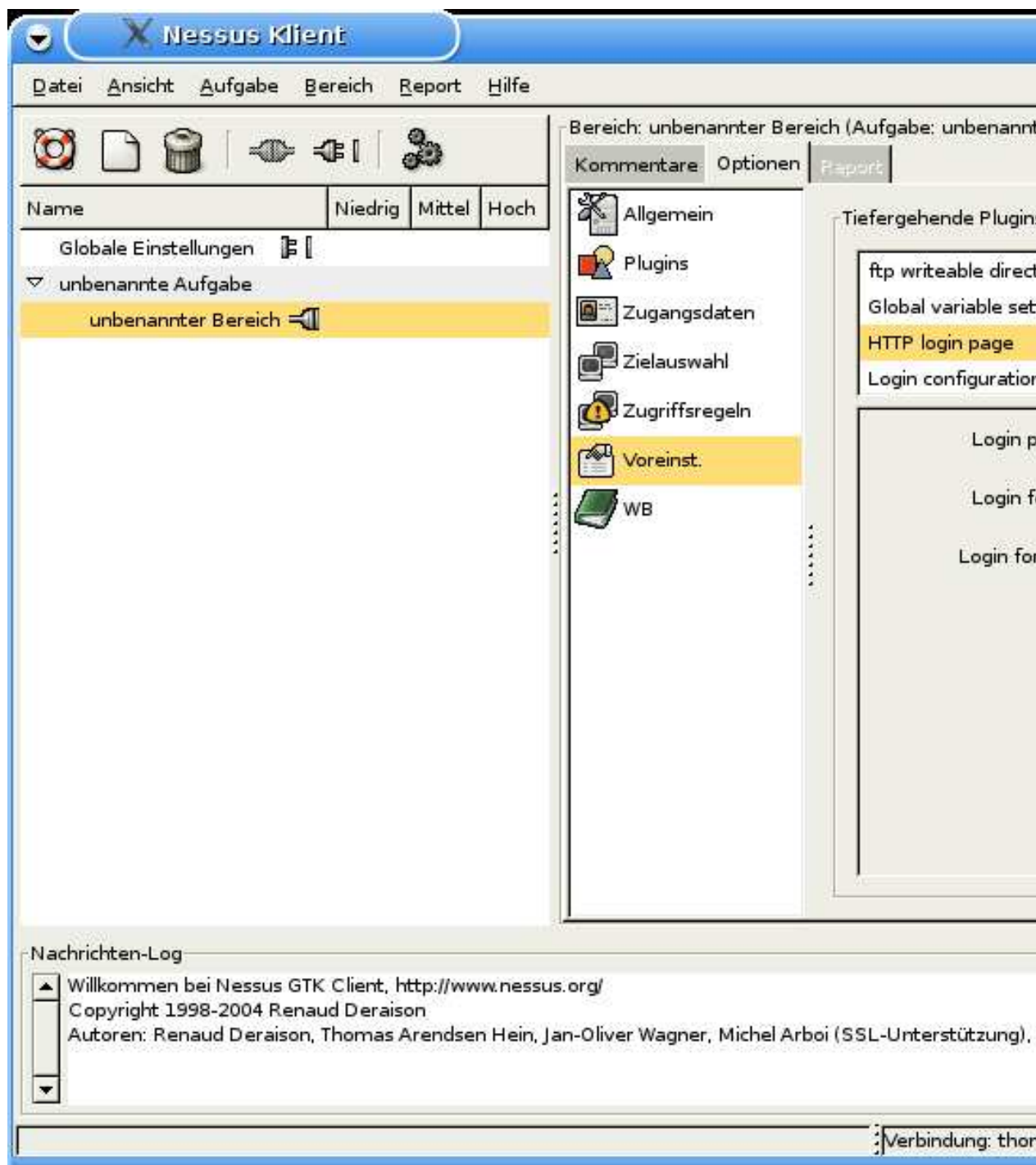
4.3.3 Zugangsdaten

Einige der Plugins bieten die Eingabe von Zugangsdaten für bestimmte Anwendungen wie z.B. Samba oder für Web-Sites (HTTP). Diese Plugins funktionieren völlig analog zu den Plugins die unter “Voreinstellungen” eine Parametereingabe erlauben. Für eine bessere Übersicht ist die entsprechende Gruppe aber unter “Zugangsdaten” getrennt aufgeführt.



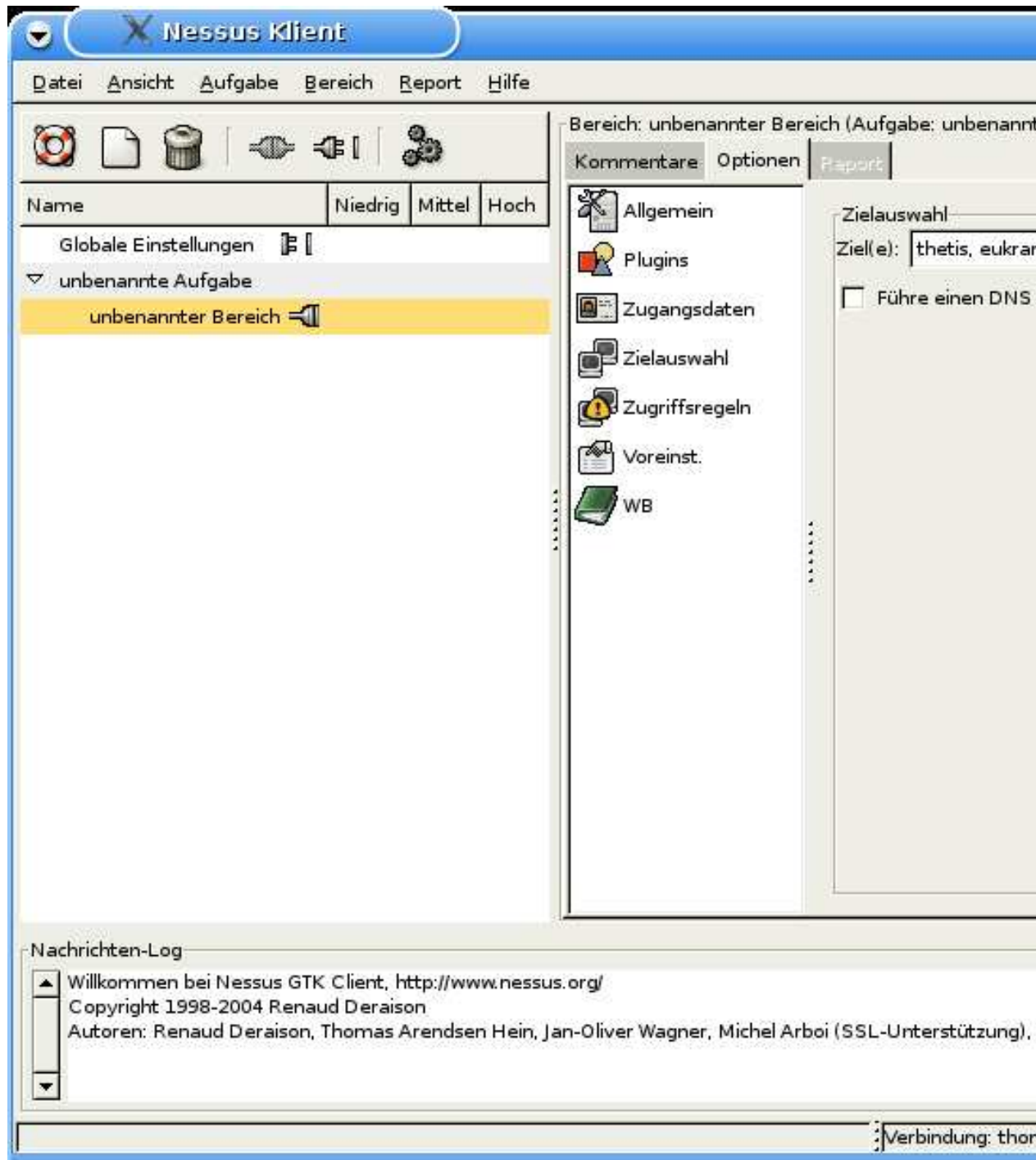
4.3.4 Plugin Voreinstellungen

Einige Plugins können anhand von Parametern angepasst werden. Die Parameter aller konfigurierbaren Plugins werden auf dieser Karte zusammengefasst und können modifiziert werden.



Nur eine vergleichsweise kleine Zahl von Plugins bietet die Möglichkeit einer Konfiguration.

4.3.5 Zielauswahl



Ziel(e) Die ersten Hosts, die durch den Nessus Server attackiert werden. Die anderen Optionen erlauben es, den Kreis der zu testenden Systeme zu erweitern. Es können diverse Primär-Ziele angegeben werden, indem sie mit Komma (,) separiert werden, z.B.: "host1,host2".

Eine besondere Syntax ist "file:/some/where/targetlist.txt": Die Liste der Ziele wird aus einer Datei gelesen. Diese kann auch mittels des Knopfs "Lese aus Datei" komfortabel ausgewählt werden, im entsprechenden Abschnitt des Handbuchs ist auch das Format der Datei beschrieben.

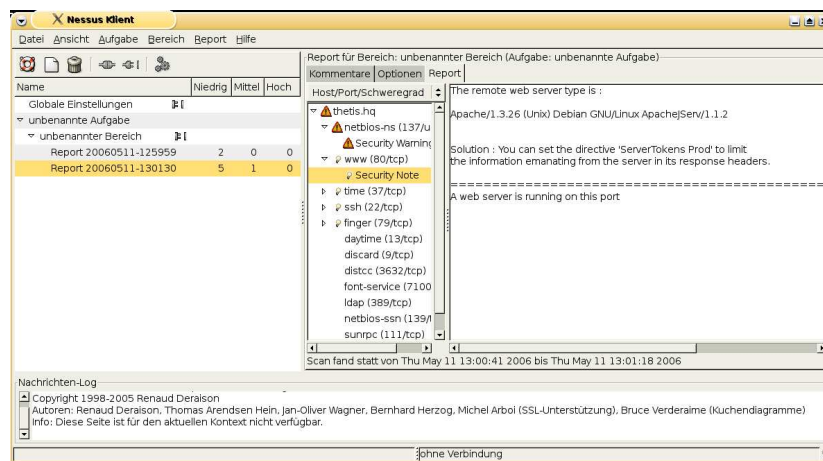
Lese aus Datei Es kann über den Dateiauswahl-Dialog eine Textdatei angegeben werden welche eine Liste der Ziele enthält. Diese Textdatei kann eine oder mehrere Zeilen mit Komma-separierten Listen von Hosts enthalten.

Führe einen DNS Zonentransfer durch Nessus Server wird AXFR Anfragen (Zonen-Transfer) an den Name-Server des Ziels richten und so eine Liste der Hosts der Ziel-Domäne ermitteln. Dann wird jeder einzelne Host getestet.

4.4 Reports

4.4.1 Report Seite von OpenVAS-Client

Die Report Seite enthält 3 Elemente: Auf der linken Seite befindet sich eine Baumstruktur welches es erlaubt über Host, Port und Schweregrad einzelne Scan-Ergebnissen zu finden. Oberhalb dieser Baumstruktur ist eine Auswahl mit der man Baumstruktur umstellen kann. Auf der rechten Seite befindet sich das Textfeld mit den eigentlichen Scan-Ergebnissen. Das gesamte Design unterstützt ein exploratives Verstehen der Scan-Ergebnisse.



4.4.2 Report Formate

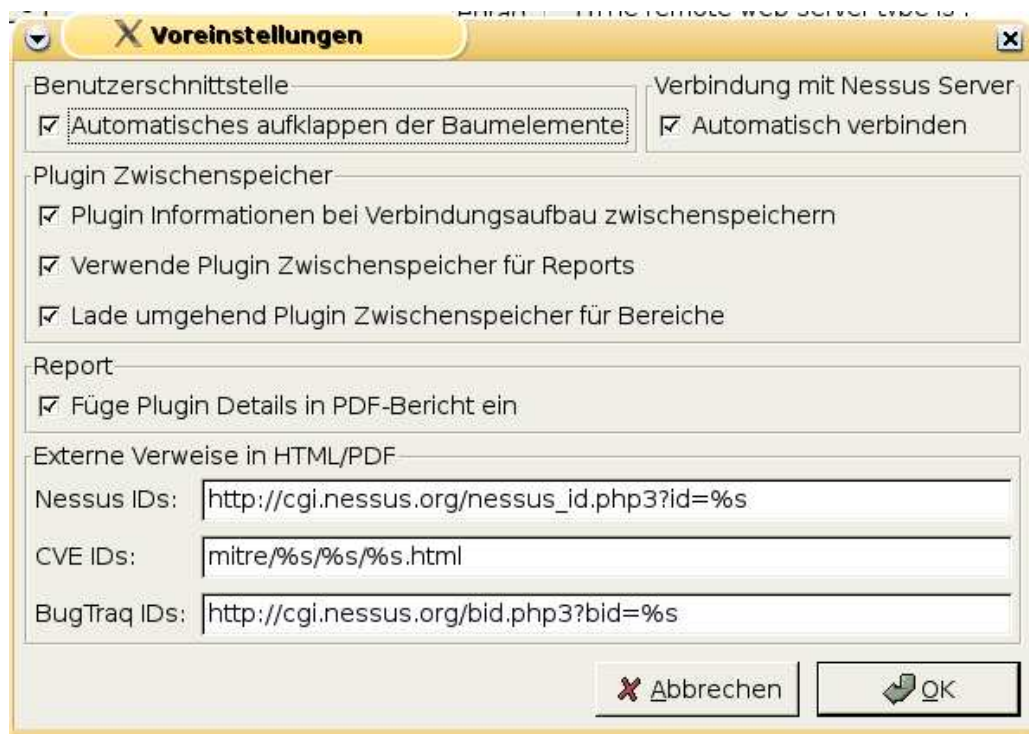
Die Scan-Ergebnisse können in verschiedene Formate exportiert werden. Grundsätzlich kann zwischen drei Formaten unterschieden werden: Austauschformate, bearbeitbare Dokumente und Read-only Dokumente. Der letzte Typ ist aktuell der PDF Report. Mit einem geeigneten Betrachter können Sie im Dokument anhand der eingebetteten Hyperlinks navigieren.

Weitere Informationen zu den Formaten finden Sie im Abschnitt über die Aktion “Report->Export”.

5 Besondere Funktionen

5.1 OpenVAS-Client Voreinstellungen

OpenVAS-Client erlaubt es, einige individuelle Vorlieben einzustellen, die das Verhalten der graphischen Benutzeroberfläche bestimmen.



Folgende Einstellungen sind verfügbar:

5.1.1 Benutzerschnittstelle

Automatisches Aufklappen der Baumelemente In der linksseitigen Baumansicht werden die Teilbäume für Aufgaben oder Bereiche jeweils automatisch aufgeklappt, wenn man sie anwählt und diese Voreinstellung eingeschaltet ist.

Ist die Option nicht ausgewählt, so wird ein Teilbaum erst dann aufgeklappt, wenn man direkt auf das entsprechende Symbol klickt.

5.1.2 Verbindung mit Nessus Server

Automatisch verbinden Ist diese Option eingeschaltet, wird OpenVAS-Client versuchen eine Verbindung zum Server aufzubauen sobald ein Bereich ausgeführt werden

soll. Bei Verwendung von Benutzer-Zertifikaten ohne Passwort funktioniert das unmittelbar. Für Passwort-geschützte Benutzer-Zertifikate oder einfache Passwort-basierte Anmeldung wird das Passwort im Speicher gehalten bis OpenVAS-Client beendet wird.

5.1.3 Plugin Zwischenspeicher

Plugin Informationen bei Verbindungsaufbau zwischenspeichern Ist dieses Option eingeschaltet, legt OpenVAS-Client für den entsprechenden Bereich einen Zwischenspeicher für die kompletten Plugin Informationen an. Das wirkt sich auf dreierlei Weise aus:

Erstens kann der erneute Verbindungsaufbau zum Nessus Server bedeutend schneller sein, da MD5 Prüfsummen verwendet werden um nur veränderte und neue Plugins zu identifizieren. Nur diese Änderungen werden dann heruntergeladen. Verbndet man sich auf einen anderen Nessus Server werden in der Regel alle Plugins wieder neu heruntergeladen.

Zweitens sind sämtliche Plugin Informationen auch dann in OpenVAS-Client verfügbar wenn man noch keine Verbindung zum Nessus Server aufgebaut hat. Man kann also ohne Server-Verbindung die Pluginauswahl und die Plugin Voreinstellungen anschauen oder ändern. Beachten Sie, dass sich die Pluginauswahl ggf. beim Verbinsaufbau ändern kann, beispielsweise wenn neue Plugins hinzugekommen sind oder welche gelöscht werden. Letzteres passiert in der Regel jedoch nie. Den Zwischenspeicher zu laden kann unter Umständen ein paar Sekunden dauern. Wollen Sie das vermeiden, dann schalten Sie die Option “Lade umgehend Plugin Zwischenspeicher für Bereiche” aus.

Drittens der Nachteil eines Zwischenspeichers: Es werden pro Bereich einige MByte an Platz benötigt. Stellt dies ein Problem dar, sollten sie diese Option abschalten. Wollen Sie bereits erzeugte Zwischenspeicher wieder entfernen, so suchen Sie nach den Dateien “nessus_plugin_cache” in Ihrem Nessus Verzeichnis “~/nessus”. Einfaches löschen dieser Dateien reicht aus.

Verwende Plugin Zwischenspeicher für Reports Schaltet man diese Option ein, so wird OpenVAS-Client sämtliche Plugin Informationen allen neuen Scan-Reports beilegen. Das erlaubt Ihnen die Pluginauswahl und die Plugin Voreinstellungen für einen Report in der grafischen Benutzeroberfläche anzuschauen. Dieser Zwischenspeicher dient also der Verbesserung der Transparenz, nicht der Performanz.

Auch hier gibt es den Nachteil, dass pro Report mehrere MByte an Platz benötigt werden. Stellt dies ein Problem dar, sollten sie diese Option abschalten. Wollen Sie bereits erzeugte Zwischenspeicher wieder entfernen, so suchen Sie nach den Dateien “nessus_plugin_cache” in Ihrem Nessus Verzeichnis “~/nessus”. Einfaches löschen dieser Dateien reicht aus.

Lade umgehend Plugin Zwischenspeicher für Bereiche Schaltet man diese Option ab, so wird OpenVAS-Client den Zwischenspeicher für einen Bereich nicht automatisch laden wenn man einen Bereich anwählt. Das bedeutet, dass man weder die Pluginauswahl noch die Plugin Einstellungen zu sehen bekommt wenn man nicht mit dem Nessus Server verbunden ist. Diese Option schaltet also den zweiten Punkte von “Plugin Informationen bei Verbindungsaufbau zwischenspeichern” wieder ab um möglicherweise störende längere Ladezeiten des Zwischenspeichers zu vermeiden wenn man einen Bereich anwählt.

5.1.4 Report

Füge Plugin Details in PDF-Bericht ein Schaltet man diese Option ein, so wird OpenVAS-Client bei der Erstellung von PDF-Reports einen Anhang mit Details zu solchen Plugins anfügen die relevante Ergebnisse des Berichts erzeugt haben. Diese Details sind innerhalb des PDF-Dokumentes verknüpft, so dass man leicht zu den gewünschten Informationen springen kann.

Bedenken Sie aber, dass das PDF-Dokument dadurch unter Umständen signifikant an Umfang gewinnen kann. Im Durchschnitt können Sie davon ausgehen, dass etwa zwei Plugin-Beschreibungen auf einer Seite stehen.

5.1.5 Externe Verweise in HTML/PDF

Diese Einstellungen bestimmen die URL für die Verweise auf weitere Informationen zu Nessus Plugins, CVE/CAN und BugTraq ID in Reports der Formate HTML und PDF. Die Voreinstellungen wie oben im Screenshot zu sehen werden empfohlen, da dort jeweils aktuelle Informationen vorhanden sind. Die Voreinstellungen erhält man zurück wenn man die entsprechenden Felder leer lässt.

Für den Fall, dass Sie einen Nessus Report als Paket mit z.B. den Details zu CVE/CAN erstellen wollen um alles off-line lesen zu können, könnten Sie diese Definition verwenden: “mitre/%s/%s/%s.html” falls Sie eine Verzeichnisstruktur relativ zur Report-Datei haben mit mitre/CVE/yyyy/nnnn.html und mitre/CAN/yyyy/nnnn.html wobei yyyy da Jahr ist und nnnn die Nummer des Eintrages. Dann können alle Dateien in einem Paket zusammengefasst werden.

Beachten Sie, dass die Zeichenketten die Sie hier definieren, in den Parameter “href” der HTML-Verweise direkt eingetragen werden. Da Werkzeug “htmldoc” wird verwendet um PDF Reports daraus herzustellen. Abhängig davon, welche Version mit welchen Eigenschaften Sie auf Ihrem System haben, kann die Art der erzeugten Verweise in der PDF-Datei variieren.

5.2 Benutzer-definierte Zugriffsregeln

5.2.1 Allgemein

Der Nessus Server verwaltet eine Liste von Zugriffsregeln ähnlich denen bei Firewall-Anwendungen. Diese Server-seitigen Regeln können durch einen Nessus Klienten niemals überschrieben werden.

Des weiteren kann der Administrator des Nessus Server zusätzlich spezielle Regeln für einen Benutzer anlegen, beispielsweise direkt bei der Einrichtung des Benutzerkontos mit `nessus-adduser`.

Schließlich hat der Benutzer noch die Möglichkeit, zusätzliche eigene Regeln in OpenVAS-Client zu setzen.

5.2.2 Syntax

Eine Regel kann zum einen eine Voreinstellungsregel sein: “default deny”, “default reject” oder “default accept”.

Zum anderen kann eine Regel Ziel-spezifisch angegeben werden und besteht dann aus einer Aktion (“deny”, “reject” oder “accept”) und dem Ziel spezifiziert durch die IP (z.B. 192.168.10.10).

Beachten Sie, dass hier tatsächlich IPs angegeben werden müssen. Hostnamen sind aus Sicherheitsgründen nicht gestattet.

5.2.3 Regeln in OpenVAS-Client verwalten

Eine Regel besteht aus einer Aktion und (falls keine voreingestellte Aktion definiert wurde) einem Ziel. Es sind insgesamt 6 Aktionen möglich aus denen der Benutzer in einer Auswahlbox auswählen kann. Das Ziel muss separat in einem Texteingabefeld angegeben werden.

Sobald man “Regel hinzufügen” anwählt, wird die spezifizierte Regel in der Gruppe “Klient-seitige Regeln” hinzugefügt.

Ist eine Regel in dieser Gruppe selektiert und betätigen Sie “Regel entfernen”, so wird die selektierte Regel aus der Liste gelöscht. Selbstverständlich können keine Server-seitigen Regeln gelöscht werden.

5.2.4 Beispiel für eine Benutzer-definierte Zugriffsregel

Sie sind das Sicherheitsteam von BigBank Corp. Das Management hat Sie autorisiert, das Netzwerk zu scannen. Allerdings unter keinen Umständen 172.20.142.2, da dies der SWIFT Server ist. Die Bank würde erheblichen Schaden erleiden, wenn dieser Server offline geht. Also fügen Sie als Klient-seitige Zugriffsregel hinzu:

```
deny 172.20.142.2
```

Sie können nun sicher sein, dass diese IP nicht gescannt wird, selbst wenn diese IP bei den Zielen explizit auftaucht.

5.2.5 Beispiel für eine Administrator-definierte Zugriffsregel

Indem Sie die Datei `nessusd.rules` editieren (nur der Administrator des Nessus Server kann dies), können Sie sicherstellen, dass der Nessus Server nicht verwendet werden kann, um bestimmte Hosts zu scannen. Falls Ihr internes Netzwerk 10.0.0.0/8 ist, dann wollen Sie in der Regel sicherstellen, dass niemand Rechner außerhalb dieses Netzes mit dem Server scannen kann. Dies erreichen Sie durch folgende Einstellung in `nessusd.rules`:

```
allow 10.0.0.0/8
allow 127.0.0.1
default deny
```

5.3 Lokale Sicherheitstests: Theorie

5.3.1 Was sind “lokale” Sicherheitstests?

Beginnend mit Nessus 2.1.0 ist es möglich, Nessus SSH-Beglaubigungen zu übergeben, um in ein anderes UNIX-artiges System einzuloggen. Dort kann dann lokal festgestellt werden, welche Patches auf dem System ggf. noch fehlen. Konkret bedeutet dies, dass Nessus einen validen SSH-Schlüssel erhält, sich damit auf den anderen Systemen einloggt, die Liste der installierten Software extrahiert und anhand dieser mitteilt, welches Softwarepaket aktualisiert werden sollte.

Das eigentliche Ziel dieser Funktionalität ist es, auf einfache Weise zu verfolgen, ob alle Patches auf den Systemen eingespielt wurden, ohne dass möglicherweise Gefahren für das System entstehen indem man aktiv auf Sicherheitslücken von Außen prüft.

Es ist also wichtig zu verstehen, dass diese Funktionalität keine versteckten Probleme wie ein unautorisiertes SUID Programm entdecken kann, sondern nur feststellt, welche Patches auf dem System fehlen.

5.3.2 Welche Betriebssysteme werden derzeit unterstützt?

Derzeit ist Nessus in der Lage fehlende Patches für folgende Betriebssysteme festzustellen:

- AIX 5.x
- Debian
- Fedora Core 1 und 2
- FreeBSD 4.x, 5.x (inklusive aller Port Sammlungen)
- Gentoo Linux (alle Empfehlungen in 2004)
- Mac OS X (und Mac OS X Server)
- Mandrake Linux (8.0 bis 10.0)
- RedHat Enterprise Linux (2.1 und 3.0)
- Solaris (2.5.1, 2.6, 7, 8 und 9)
- SuSE Linux (7.0 bis 9.1)
- Microsoft Windows NT, 2000, XP, 2003

Wir hoffen, in der Zukunft weitere Systeme (HP-UX, Tru64, etc...) unterstützen zu können.

5.3.3 Nach welchen Solaris Patches wird geprüft?

Es sind Prüfungen für Solaris 2.5.1, 7, 8 und 9 (für beide Architekturen: sparc und i386) implementiert. Wir prüfen auf jeden Patch der auf der offiziellen Sun Patch-Seite "Patch containing security fixes" aufgelistet wird.

5.3.4 Ist es geplant weitere Systeme zu unterstützen?

Es ist nicht geplant, alle jemals erstellten Betriebssysteme zu unterstützen. Allerdings haben wir vor, "Arbeitsgruppen" für bestimmte Betriebssysteme zu schaffen. Im Prinzip suchen wir also nach Menschen, die für das von Ihnen bevorzugte Betriebssystem verantwortungsbewußt sämtliche notwendigen Prüfungen auf Patches implementieren. Falls Sie dementsprechend interessiert sind, so melden Sie sich am besten beim Nessus Entwickler-Team.

5.3.5 Was sind lokale Sicherheitsprüfungen NICHT?

Nessus selbst prüft nicht die lokalen Sicherheitsregeln ab, wie es beispielsweise TIGER macht (also Verzeichnis mit Schreibrechten für alle, falsche Zugriffsrechte bei sensitiven Konfigurationsdateien, etc). Lesen Sie bei SLAD (Security Local Auditing Daemon) nach, wie man umfangreiche Vor-Ort-Tests auf einem potenziell kompromitierten System durchführt. SLAD kann über Nessus mit den entsprechenden NASL Skripten verwendet werden.

5.3.6 Warum benötigt man lokale Sicherheitstests, ist Nessus nicht gut genug, die Sicherheitsprobleme selbst zu entdecken?

Nessus 2.0 ist im Kern ein Netzwerk-basierter Scanner. Das bedeutet, dass er sich mit den Zielsystemen verbindet und versucht zu beurteilen ob irgendetwas die Dienste negativ beeinflusst. Nun hat aber diese Methode, alles über das Netzwerk zu prüfen, auch einige Nachteile. Insbesondere kann ein 100% Netzwerk-basierter Scanner folgende Dinge nicht leisten:

- Klient-seitige Fehler feststellen: Es gibt heutzutage mehr und mehr Klient-seitige Verletzbarkeiten, die von Fehlern in Mail-Programmen bis zu XML Bibliotheken reichen. Es ist schlichtweg unmöglich, auf diese Probleme hin mit einem Scanner wie Nessus zu prüfen. Die einzige Möglichkeit dies zu tun ist über einen passiven Scanner (z.B. SLAD oder NeVO) oder eben lokale Sicherheitstests auf Patches.
- Lokale Fehler feststellen: Klar ist, dass die Sicherheit der lokalen Werkzeuge nicht über das Netzwerk beurteilt werden kann. Indem man aber einen lokalen Sicherheitstest laufen läßt, kann man feststellen ob man einen Overflow in einem SUID-Programm ausnutzen kann, um die Rechte von root zu erlangen.
- Fehler in abgestellten Diensten feststellen: Hat ein Host einen verletzbaren Dienst installiert, aber dieser läuft nicht zur Zeit des Netzwerk-Scans, so wird diese Verletzbarkeit nicht entdeckt. Hat man einen ungepatchten Dienst mit einem Sicherheitsloch, so ist dies ein echtes Sicherheitsproblem, denn die Lücke kann sofort ausgenutzt werden, sobald der Dienst in Betrieb genommen wird.

Die traditionellen Netzwerk-basierten Scanner haben also ihre Grenzen, die aber mit der neuen Funktionalität überwunden werden können.

5.3.7 Kann Nessus individuell compilierte und installierte Pakete erkennen?

Falls Sie beispielsweise Ihren DNS Server per Hand installieren und anstatt des System-seitigen verwenden, so wird Nessus dies nicht erkennen!

Nessus beschränkt sich auf die Patches, die durch den Betriebssystemhersteller zur Verfügung gestellt werden. Beachten Sie, dass es normalerweise eine gute Idee ist, bei den Paketen des Herstellers zu bleiben, da dies eine Aktualisierung des Systems wesentlich vereinfacht.

In der Zukunft wird Nessus in der Lage sein, zu prüfen, ob der laufende Dienst derselbe ist wie das installierte RPM (oder anderen Paket-Management-Systems) und wird so die Ausgabe-Informationen noch verbessern.

5.3.8 Warum um lokale Sicherheitstests kümmern, wenn es gar keine lokalen Benutzer gibt?

Warum sollte es beispielsweise jemanden kümmern, ob der Web-Server einen lokalen Overflow in einem SUID-Programm hat, wenn es keinen lokalen Benutzer außer einen selbst gibt?

Als diese Funktionalität diskutiert wurde, haben eine Leute entgegengehalten, dass es Ihnen egal ist, da ihre Server einfach nur Web-Server ohne lokale Benutzer sind und daher niemand die Verletzbarkeiten ausnutzen kann. Die Antwort darauf ist eine einfache Praxis-Regel der Sicherheit: **Eindämmung**. Wenn Ihr Web-Server verletzbar durch einen bestimmten Fehler ist (sei es eine fehlerhafte Konfiguration, ein schlecht

geschriebenes PHP-Skript oder ein Oday in Apache), dann kann ein Angreifer eine Shell auf Ihrem System erhalten. Nun ist die Frage: Wollen Sie es diesem Angreifer einfach machen, Administrator-Privilegien zu erhalten und damit die vollständige Kontrolle über das System? Oder wollen Sie ihn eindämmen auf die Privilegien des Web-Servers? (Diese Privilegien werden zumindest verhindern, dass der Angreifer den Kern verändert, gefährliche Kern-Module lädt und ganz allgemein es schwieriger macht überhaupt zu entdecken, dass eine Attacke erfolgreich war). Also, lokale Sicherheit ist wirklich wichtig, denn sie ist die letzte Verteidigungslinie.

5.3.9 Wie kann das Einschalten der lokalen Sicherheitstests die Scan-Erfahrung verbessern?

Abhängig davon, wie Sie sie benutzen, können lokale Sicherheitstests Ihre Scan-Erfahrung dramatisch verbessern:

- Durch Reduzierung der falschen Positiven: es ist nicht immer möglich über das Netzwerk festzustellen, ob ein Dienst für einen bestimmten Fehler verletzbar ist (manchmal ist die einzige Möglichkeit dies zu tun, den Dienst zum Absturz zu bringen, was natürlich keine Option für Produktiv-Systeme ist). Manchmal aktualisieren Hersteller ein fehlerhaftes Produkt nicht, sondern wenden den Patch auf die alte Version an (siehe z.B. den Standpunkt von RedHat dazu). Dies macht den Netzwerk-basierten Scannern natürlich das Leben etwas schwerer.
- Durch Beschleunigung der Prüfungen: wenn Sie eigentlich nur daran interessiert sind zu prüfen ob auf allen Systemen alle Patches eingespielt sind, dann wird die Verwendung der lokalen Sicherheitstests die Zeit für die Prüfung drastisch reduzieren. Es dauert nur 6 Sekunden alle lokalen Fehler zu finden (über einen VPN-Link) während es mindestens 1 Minute dauert, das selbe System über das Netzwerk zu scannen.
- Durch häufigeres Scannen: Jeder Netzwerk-basierte Sicherheits-Scanner birgt die Gefahr, Geräte oder Dienste auf dem Netzwerk abstürzen zu lassen. Das gilt für Nessus wie für alle anderen vergleichbaren Systeme. Der Grund ist einfach: Ein Scanner versucht sich zu allen Diensten auf dem Zielrechner zu verbinden und einige dieser Dienste (und TCP/IP Stacks) sind so schlecht geschrieben, dass, wenn man nicht genau die richtigen Daten schickt, ein Absturz die Folge ist. Egal wie vorsichtig der Scanner ist, es gibt immer die Gefahr spezielle Anwendungen zum Absturz zu bringen oder einen belasteten Server mit wenig Speicherreserve usw. Dies ist im Speziellen der Fall für Embedded-Geräte, die meistens ganz ohne Überlegungen zur Sicherheit konstruiert werden. Es gibt Hersteller von Scannern die "100% ungefährliche Scans" versprechen, was schlichtweg eine Lüge ist. Interagiert ein Scanner mit Netzwerkdiensten, so gibt es immer die Möglichkeit, dass diese zum Absturz gebracht werden. In der Konsequenz wird nur noch einmal pro Monat oder einmal pro Quartal gescannt, obwohl neue Sicherheitsprobleme jeden Tag hinzukommen bzw. entdeckt werden. Alleine dadurch, dass die lokalen Sicherheitstest eingeschaltet werden und die Port-Scanner sowie alle anderen Plugins ausgeschaltet werden, wird Nessus Server sich nur noch per SSH auf die Zielrechner einloggen und keinen Stress auf dem Netzwerk erzeugen und auch nicht zu schlechten Überraschungen führen.

Es ist also egal, ob sie eine kleine oder sehr große Anzahl Systeme haben, lokale Sicherheitstests machen Ihnen Ihre Arbeit in jedem Fall einfacher.

5.3.10 Verlangsamt diese Funktionalität den Scan?

Nein. Nessus minimiert die Anzahl der SSH Zugriffe auf die Zielrechner und loggt sich tatsächlich nur ein einziges mal pro System ein, um die Daten dann in die Wissensbasis zu übertragen. Schauen Sie sich den Quelltext des Plugins `ssh_get_info.nasl` für Details dazu an.

5.3.11 Was sind die Nachteile dieser Funktionalität?

Derzeit sind die folgenden Einschränkungen bekannt:

- Es wird nur SSHv2 unterstützt. Ein Zugriff auf Hosts über das Protokoll SSHv1 ist nicht möglich.
- Sie müssen die Authentifizierungsmethode über öffentliche Schlüssel mit SSH verwenden. Es ist zwar etwas komplizierter als einfache Passworte, dafür aber um so sicherer.
- Diese Dokumentation beschreibt nur OpenSSH.

5.4 Lokale Sicherheitstests: Wie man sie anschaltet

5.4.1 Das Prinzip

Für das Aktivieren von lokalen Sicherheitstests liegt die folgende Idee zugrunde:

- Erzeuge ein SSH Schlüsselpaar mit öffentlichem und privatem Schlüssel, die dann von Nessus verwendet werden.
- Erzeuge ein Benutzerkonto auf jedem System auf dem die lokalen Sicherheitstests durchgeführt werden sollen.
- Kopiere den öffentlichen SSH Schlüssel den Nessus verwendet in das Verzeichnis der jeweiligen Benutzer.
- Weise Nessus an, diese SSH Schlüssel zu verwenden um die Scans durchzuführen.

Wie man sieht ist es also eine einfache Methode. Nessus verwendet keine direkte Authentifizierung über Passwort, da sonst eine Verletzbarkeit über eine man-in-the-middle Attacke gegeben wäre (derzeit wollen wir das Schlüsselmanagement für die Zielsysteme nicht selbst übernehmen)

Wenn Sie sich nicht mit SSH auskennen und nicht verstehen was Schlüsselpaare mit öffentlichem und privatem Schlüssel sind, sollten Sie unbedingt zunächst entsprechende Dokumentation und Literatur heranziehen.

5.4.2 Einen öffentlichen SSH-Schlüssel erzeugen

Um einen SSH Schlüssel zu erzeugen, verwenden Sie das Kommando `ssh-keygen` und speichern Sie den geheimen Schlüssel an einem sicheren Ort:

```
$ ssh-keygen -t dsa
Generating public/private dsa key pair. Enter file in which to save the key
(/Users/renaud/.ssh/id_dsa):
/home/renaud/Nessus/ssh_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/renaud/Nessus/ssh_key.
Your public key has been saved in /home/renaud/Nessus/ssh_key.pub.
The key fingerprint is: 06:4a:fd:76:ee:0f:d4:e6:4b:74:84:9a:99:e6:12:ea
renaud@myth.local
```

Wie Sie am Beispiel oben sehen können, wurden zwei Dateien erzeugt:

- `/home/renaud/Nessus/ssh_key` ist der private SSH Schlüssel. Transferieren Sie ihn nie auf ein anderes System als das auf dem der Nessus Klient läuft.
- `/home/renaud/Nessus/ssh_key.pub` ist der öffentliche SSH Schlüssel. Diese Datei wird auf jedes System kopiert für welches lokale Sicherheitstests mit Nessus gemacht werden sollen.

Wenn Sie bei `ssh-keygen` nach einem Passwort gefragt werden, dann können Sie auch zweimal mit Return bestätigen (also kein Passwort spezifizieren). Natürlich ist es besser einen Schlüssel mit einem Passwort zu schützen. Aber in diesem Fall ist es so, dass keine zusätzliche Sicherheit gewonnen wird, da das Passwort zwischen Nessus Klient und Nessus Server zirkuliert und zudem in der Datei `.nessusrc` im Klartext gespeichert wird.

5.4.3 Ein Benutzerkonto einrichten und einen SSH Schlüssel herstellen

Für jede Plattform die Sie testen wollen, sollten Sie ein neues Benutzerkonto einrichten, welches nur für die Verwendung durch Nessus vorgesehen ist. In diesem Dokument verwenden wir den Namen `nessus`, das Benutzerkonto könnte aber auch jeden anderen Namen haben. Nachdem das Benutzerkonto eingerichtet wurde (mit dem Kommando `adduser` oder einem entsprechenden Werkzeug welches Ihr System bereitstellt), sollten Sie sicherstellen, dass dieser neue Benutzer kein gültiges Passwort hat (also einen Stern (*) an der entsprechenden Stelle in der Passwort-Datei):

```
# vipw
(...)
nessus:*:502:502::0:0:Nessus Test Account:/home/nessus:/bin/bash
(...)
```

Da der Benutzer nun eingerichtet ist, müssen Sie nur noch den Schlüssel in dessen Heimatverzeichnis kopieren und die Zugriffsrechte für das Verzeichnis `“ssh”` korrekt setzen:

```
# mkdir ~nessus/.ssh
# mv ssh_key.pub ~nessus/.ssh/authorized_keys
# chown -R nessus:nessus ~nessus/.ssh/
# chmod 0600 ~nessus/.ssh/authorized_keys
# chmod 0700 ~nessus/.ssh/
```

5.4.4 Nessus aufsetzen

Stellen Sie zunächst sicher, dass Sie einen Nessus Server in Version 2.1.0 oder neuer haben, der mit SSL Unterstützung compiliert ist. Sie können dies prüfen mit “nessusd -d”:

```
[root@myth Home]$ nessusd -d
This is Nessus 2.1.0 for Darwin 7.4.0
compiled with gcc version 3.3 20030304 (Apple Computer, Inc. build 1495)
Current setup :
nasl: 2.1.0
libnessus: 2.1.0 SSL support: enabled
```

Starten Sie nun OpenVAS-Client und gehen Sie zur Seite mit den Zugangsdaten und setzen Sie den SSH Benutzernamen und den SSH Schlüssel:

Stellen Sie dabei sicher, dass Sie nicht den öffentlichen Schlüssel (.pub) und den privaten vertauschen. Stellen Sie außerdem sicher, dass die Option “Abhängigkeiten zur Laufzeit berücksichtigen” eingeschaltet ist und starten Sie dann den Scan gegen den Host. Der Nessus Report sollte nun mitteilen, dass das Einloggen erfolgreich war und alle fehlenden Patches für dieses System auflisten:

Wiederholen Sie diese Prozedur für jeden Rechner, für den Sie eine lokale Prüfung machen wollen.

5.4.5 Danksagung

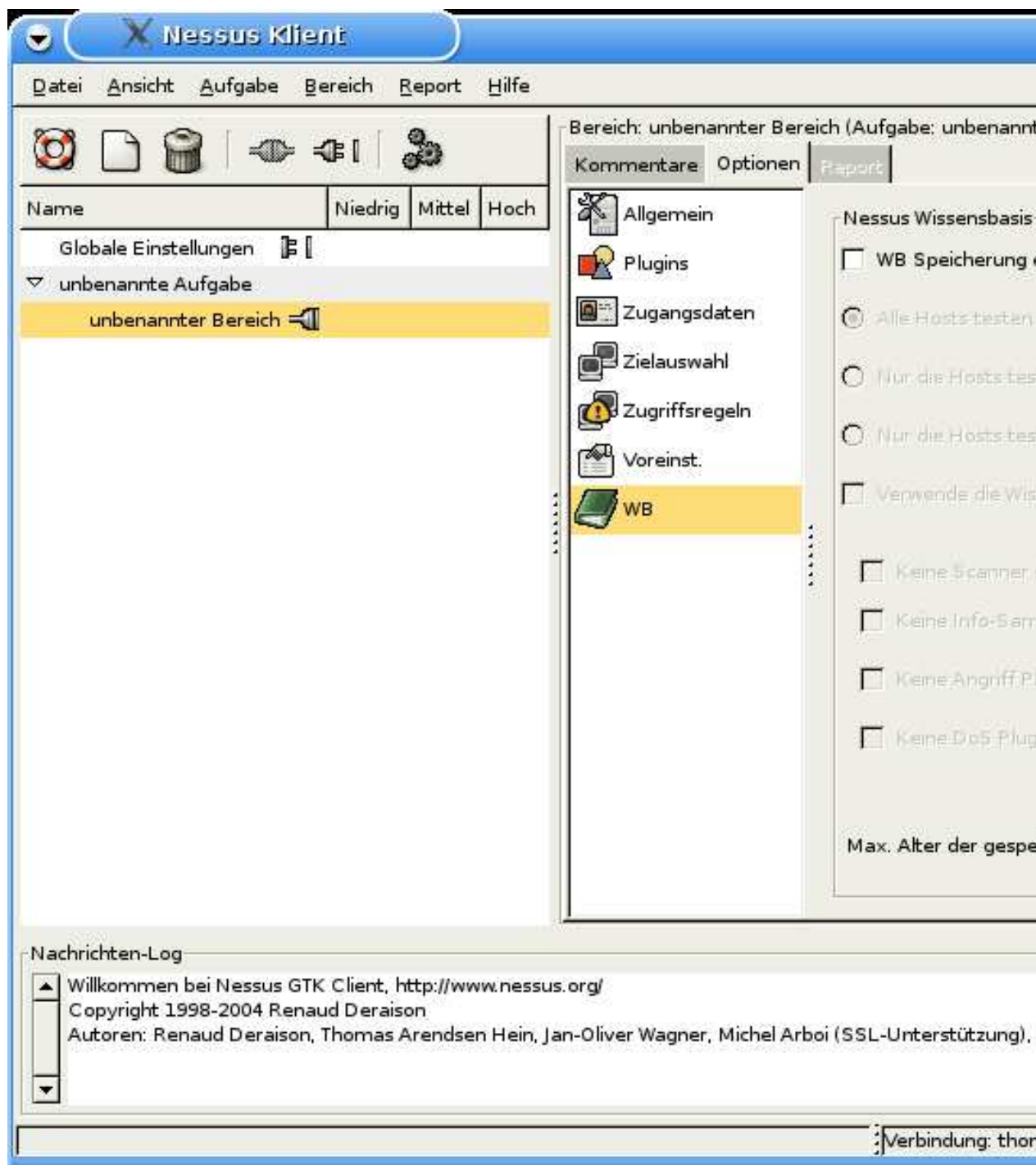
Die Funktionalität der SSH-basierten lokalen Tests wurde beigesteuert durch:

- Nicolas Pouvesle, der einen leichten SSH Klienten in NASL entwickelt hat.
- Tenable Network Security, welche über 1200 Plugins für lokale Sicherheitstests hergestellt haben.

6 Optimierung und Feinabstimmung

6.1 Wissensbasis

Die Speicherung in der Wissensbasis erlaubt es, die Benutzer des Netzwerkes weniger zu stören indem tägliche Scans eines /24 Netzwerkes durchgeführt werden, ohne dass dabei die Testresultate jedes mal wieder verworfen werden.



Sowohl der Nessus Server, als auch der Nessus Klient, haben eine Compiler-Option, um die Unterstützung durch eine Wissensbasis prinzipiell zu integrieren. Dies ist auch voreingestellt, aber man kann auf Nessus Server treffen, die die Wissensbasis nicht unterstützen.

Im OpenVAS-Client finden Sie die Konfiguration für die Wissensbasis auf der Seite “WB” der Optionen (falls dieser Klient entsprechend compiliert wurde).

6.1.1 Einführung

Die Wissensbasis (WB) ist die Liste der Informationen, die für einen getesteten Host zusammengetragen wurden. Diese beinhaltet die Liste der geöffneten Ports, den Typ des Betriebssystems und eine Reihe weiterer Informationen. Die Hauptintention ist die Reduzierung der Redundanz bei Tests, so dass ein Plugin A, welches ein Fakt findet (z.B. einen Weg wie man in den FTP-Server einloggt) diese Information mit anderen Plugins teilen kann (beispielsweise wird ein Plugin, welches für seinen Test in einen FTP-Server einloggen muss, in der WB nachschauen, ob es einen Weg gibt). Nach einem Test wird die WB aus dem Speicher entfernt und beim nächsten Test von Grund auf neu zusammengetragen.

Die Idee hinter der Speicherung der Wissensbasis ist es, die Ergebnisse erneut auch für andere Audits zu verwenden, um damit die Bandbreite zu schonen. Hat man beispielsweise die Konfiguration seines Web-Servers in den letzten 5 Stunden nicht verändert, dann ist es recht unwahrscheinlich, dass dort ein neuer Port geöffnet wurde, der bei einem erneuten Scan berücksichtigt werden sollte. Die Speicherung der Wissensbasis erlaubt:

- Schonung der Bandbreite
- Zeitersparnis
- Test der Hosts nur auf neue Verletzlichkeiten

Man kann eine Lebenszeit für die Wissensbasis für jeden Host definieren. Diese Lebenszeit kann frei gewählt werden und zwischen einigen Sekunden und ein paar hundert Jahren liegen. Dieser Abschnitt erklärt wie man die Wissensbasis verwendet um einen Sicherheitszugewinn durch tägliche Scans ganzer Netze zu erreichen.

6.1.2 Verwendung der WB Speicherung

Wollen Sie die Wissensbasis Server-seitig speichern lassen, so müssen Sie die Option “**WB Speicherung einschalten**” im Klienten einschalten:

Diese Option aktiviert die Speicherung der Wissensbasis. Auf dem Nessus Server werden alle entsprechenden Informationen gesammelt und gespeichert. Falls diese Option ausschließlich eingeschaltet ist, so werden die Daten lediglich gespeichert, aber eben nicht weiter verwendet. Das bedeutet, dass der Nessus Server diese Daten nicht nutzen wird, wenn er denselben Host erneut testet.

Technisch gesehen wird jede Wissensbasis als separate Datei für jeden Host auf dem Nessus Server gespeichert. In Zukunft ist die Verwendung einer Datenbank denkbar. Die Dateien liegen unter “/usr/local/var/nessus/username/kbs/”, wobei username das Benutzerkonto ist unter welchem man entsprechend gearbeitet hat.

6.1.3 Reduzierung der getesteten Hosts in der Wissensbasis

Falls “**WB Speicherung einschalten**” gesetzt ist, dann ist es auch möglich, die Menge der getesteten Hosts die dort abgelegt werden für den jeweiligen Benutzer einzuschränken. Unter Umständen wollen Sie nur diejenigen Hosts testen, die bereits in der

Wissensbasis vorliegen. Andersherum wollen Sie ggf. nur die Hosts testen, für die keine oder eine veraltete Wissensbasis vorliegt, Sie damit also den Satz der Wissensbasis komplett und aktuell machen. Die drei Optionen **“Alle Hosts testen”**, **“Nur die Hosts testen welche bisher getestet wurden”** und **“Nur die Hosts verwenden die bisher nicht getestet wurden”** erlauben die entsprechenden Einstellungen.

- Falls **“Alle Hosts testen”** eingestellt ist, dann werden sämtliche Hosts, die bei **“Ziele”** stehen, getestet. Es wird also keine Filterung gemacht und ein Test wie gewohnt durchgeführt.
- Wenn Sie die Option **“Nur die Hosts testen welche bisher getestet wurden”** auswählen, dann werden nur diejenigen Hosts getestet, für die eine aktuelle Wissensbasis vorliegt. Dies erlaubt es, beispielsweise darin sicher zu gehen, dass kein Port-Scan gegen einen Host durchgeführt wird der noch nicht getestet wurde aber in der Ziel-Liste steht.
- Schließlich, falls die Option **“Nur die Hosts verwenden die bisher nicht getestet wurden”** gesetzt ist, werden nur diejenigen Hosts getestet, für die keine Wissensbasis vorliegt oder der Wissensbasis veraltet ist. Diese Option sorgt also für eine frische Wissensbasis für entsprechende Hosts.

Die drei Optionen erlauben folgendes:

- Alles testen
- Sicherstellen, dass kein störendes Rauschen während der Tests entsteht
- Die Wissensbasis zusammenzustellen

6.1.4 Wiederverwendung der Wissensbasis für Mehrfachtests

Die Wiederverwendung der Wissensbasis, die in der Vergangenheit gespeichert wurde, kann auch gefährvoll sein, beispielsweise wenn ein gestern getestetes System über Nacht kompromittiert und mit einem Root-Kit versehen wurde und der Eindringling eine Root-Shell auf einem Port stehen läßt der vorher geschlossen war. Wenn Sie die Wissensbasis wiederverwenden so werden Sie diesen Fall nicht mitbekommen. Auf der anderen Seite will man auf einem gut abgesicherten Netzwerk die Wissensbasis wiederverwenden, um Zeit zu sparen, wenn man einen Satz neuer Tests gegen einen Host fahren lassen möchte.

Wenn Sie die Option **“Verwende die Wissensbasis für die Hosts beim Test”** aktivieren, dann werden die vorherigen Wissensbasen in den Speicher geladen und Nessus Server wird sie aktiv verwenden. Sie können mit den folgenden vier Optionen kontrollieren, was der Nessus Server genau machen soll:

Ist **“Keine Scanner starten die schon einmal gestartet wurden”** gesetzt, dann werden die Scanner die bereits vorher einmal verwendet wurden nicht erneut verwendet. Analog funktionieren die drei anderen Optionen. Falls Sie nur die neuen Tests laufen lassen wollen die Sie über das Kommando **“nessus-update-plugins”** erhalten haben, so dann sollten Sie alle vier Optionen einschalten.

6.1.5 Bedeutung von aktuell

In diesem Abschnitt wurde davon gesprochen, dass die Wissensbasis aktuell sein kann oder veraltet. Die Bedeutung dafür ist frei konfigurierbar. Jedes mal, wenn eine Wissensbasis erstellt wird, so wird ein Zeitstempel mitgespeichert. Die Option “**Max. Alter der gespeicherten Wissensbasis (in Sek.)**” definiert die Zeit nach der eine Wissensbasis als obsolet angesehen wird und durch eine aktuellere überschrieben wird.

Die Voreinstellung ist 864000 (10 Tage), aber Sie sind frei dies beispielsweise auf 3600 (eine Stunde) zu senken bzw. auf jeden beliebigen anderen Wert zu setzen.