

Integration von Verwundbarkeits-Analyse und Netzwerk-Management mittels Vulture *

Ulrich Flegel, Marcus Kommnick

Universität Dortmund

D-44221 Dortmund

{flegel, kommnick}@ls6.cs.uni-dortmund.de

1 Verwundbarkeits-Analyse

Die Durchsetzung von Schutzzielen hinsichtlich IT-Infrastrukturen ist in der Regel nicht mit der Installation von Schutzmaßnahmen abgeschlossen. Vielmehr befinden sich die Verantwortlichen in einem kontinuierlichem Absicherungsprozeß, bedingt durch die Dynamik heutiger IT-Infrastrukturen. Neue Verwundbarkeiten entstehen etwa durch Veränderungen der Netztopologie oder Systemkonfigurationen, durch hinzugefügte Systeme bzw. Software. Andere Verwundbarkeiten schlummern in bereits vorhandener Software, bis jemand einen Weg findet, diese auszunutzen.

Vorhandene Verwundbarkeiten können von Angreifern ausgenutzt werden. Folgenlose Angriffe bleiben häufig unbemerkt, oft aber auch erfolgreiche Einbrüche. Hier können Intrusion-Detection-Systeme (IDS) die Entdeckung des tatsächlich (versuchten) Ausnutzens von Verwundbarkeiten melden. Das Nachvollziehen von Angriffen gibt Hinweise auf vorhandene Verwundbarkeiten, die es zu beseitigen gilt. Diese reaktive Vorgehensweise überläßt jedoch dem Angreifer den ersten Schritt, der für ihn bereits zum Erfolg führen kann.

Es empfiehlt sich deshalb zusätzlich eine proaktive Herangehensweise zum Aufspüren vorhandener Verwundbarkeiten. CERT-Warmmeldungen bieten z.B. Informationen zu aktuellen Software-Verwundbarkeiten. Informationen zu Verwundbarkeiten durch Fehlkonfigurationen oder durch neue Systeme bzw. Topologieveränderungen erhält man aber erst durch die Inspektion der Systeme und Netze. Das regelmäßige manuelle gründliche Aufspüren von Verwundbarkeiten ist eine sehr aufwendige und auch fehlerträchtige Aufgabe.

*Die beschriebenen Arbeiten werden derzeit zum Teil von der Deutschen Forschungsgemeinschaft gefördert unter Bi 311/10-2.

1.1 Vorhandene Werkzeuge

Die meisten Tests zum Aufspüren von Verwundbarkeiten lassen sich automatisieren. Entsprechende Analyse-Werkzeuge sind sowohl frei als auch kommerziell verfügbar.

Viele Verwundbarkeiten werden in der Untergrund-Szene aufgedeckt. Von dort stammen dann als Nachweis häufig die entsprechenden Werkzeuge, die allerdings meist auf die berichtete Verwundbarkeit spezialisiert sind. Diese Werkzeuge verfügen in der Regel über keine graphische Nutzeroberfläche. Einen Überblick findet man beispielsweise unter Exploit World [1].

Diesem Mangel widmen sich umfassendere Analyse-Werkzeuge wie etwa Nessus [2], die versuchen, möglichst viele Verwundbarkeiten abzudecken und die gleichzeitig eine graphische Nutzeroberfläche mit Reporting bieten. Die Oberflächen der einzelnen Analyse-Werkzeuge unterscheiden sich allerdings stark von einander, insbesondere aber von den Oberflächen der üblichen Netzwerk-Management-Werkzeuge. De facto werden einzelne mittels Analyse-Werkzeugen aufgespürte Verwundbarkeiten den Administrator veranlassen, die Konfiguration der Infrastruktur mit Hilfe von Netzwerk-Management-Werkzeugen anzupassen. Dabei muß er mit mehreren Oberflächen hantieren und muß Ergebnisse der Verwundbarkeits-Analyse-Werkzeuge manuell im Netzwerk-Management umsetzen.

Der Administrator sieht sich dadurch einerseits mit sehr aktuellen aber Kommandozeilen-orientierten Einzelwerkzeugen und andererseits mit einer Vielzahl verschiedener graphischer Nutzeroberflächen konfrontiert. Es entsteht ein erheblicher Konfigurationsaufwand schon allein dadurch, daß der Administrator die IT-Infrastruktur für die einzelnen Oberflächen jeweils spezifisch modellieren und eingeben muß.

Dementsprechend läßt sich die Situation des Administrators verbessern, indem Verwundbarkeits-Analyse und Netzwerk-Management integriert werden. Der Beitrag unseres Projekts *Vulture* ist die Bereitstellung einer integrierten Plattform für Netzwerk-Management und Verwundbarkeits-Analyse, auf deren Basis sich inhaltliche Querbezüge zwischen beiden Welten automatisieren lassen.

2 Integrations-Vision

Aus Administratoren-Sicht ist die Gewährleistung der Sicherheit einer IT-Infrastruktur lediglich ein Teilbereich des Gesamtkomplexes Netzwerk-Management. Für letzteres stehen bereits mächtige Werkzeuge zur Verfügung, wie etwa OpenView [3], Tivoli [4] und Scotty [5]. Diese Werkzeuge verfügen nach entsprechender Konfiguration bereits über ein Modell der zu verwaltenden IT-Infrastruktur und operieren in vielfältiger Weise darauf.

Die Integration von Verwundbarkeits-Analyse und Netzwerk-Management bietet offensichtlich ein erhebliches Synergie-Potential hinsichtlich Effizienz und Effektivität. Administratoren arbeiten nur noch mit einer abgestimmten Oberfläche. Auch wird nur noch ein umfassendes Infrastruktur-Modell benötigt, das zusätzlich Facetten der Sicherheit beinhaltet. Inhaltlich vor-

handene Querbezüge zwischen Verwundbarkeits-Analyse und Netzwerk-Management werden nutzbar gemacht und sind damit auch automatisierbar.

2.1 Beispiel-Szenarien

An dieser Stelle seien knapp einige Beispielszenarien skizziert, die die Nutzung von Querbezügen zwischen Verwundbarkeits-Analyse und Netzwerk-Management illustrieren. Zum einen können Änderungen am Modell oder der Konfiguration der IT-Infrastruktur eine neue Verwundbarkeits-Analyse notwendig machen. Zum anderen können die Ergebnisse einer Verwundbarkeits-Analyse Anpassungen der IT-Infrastruktur erfordern.

Konfiguration → **Analyse:** Dem Infrastruktur-Modell wurden neue Rechner und eine Fallback-Route hinzugefügt. Außerdem wurden die Filterregeln eines Routers modifiziert. Das Management-System kann vorschlagen, die hinzugefügten Rechner auf Verwundbarkeiten hin zu analysieren, zu ermitteln, ob über die Fallback-Route Systeme ungewollt erreicht werden können, und ob die neuen Filterregeln verwundbare Netzwerk-Dienste verfügbar machen.

Analyse → **Konfiguration:** Eine regelmäßige Verwundbarkeitsanalyse hat Rechner aufgespürt, die offenbar inoffiziell in Betrieb genommen wurden. Außerdem wurden auf diversen internen Maschinen verwundbare Versionen von Netzwerkdiensten entdeckt. Das integrierte Management-System kann nun von sich aus Vorschläge machen, die nach Bestätigung automatisch umgesetzt werden. Die inoffiziellen Rechner werden in ihrem Netzwerk isoliert, indem die Router entsprechende Filterregeln erhalten. Auf demselben Weg können die Ports der verwundbaren Dienste gesperrt werden.

3 *Vulture*

Das Projekt *Vulture* bietet ein Rahmenwerk zur Integration von Verwundbarkeits-Analyse und SNMP-basiertem Netzwerk-Management. Dabei werden die Management-Stationen und die Agenten unterschieden, die die Verwundbarkeits-Analyse durchführen (s. Abb. 1).

Die *Vulture*-Agenten bieten eine eigenständige und portierbare Plattform zur Durchführung von Verwundbarkeits-Analysen. Die Kommunikation mit *Vulture*-Agenten geschieht mittels SNMP. Die *Vulture*-Agenten sind mit Hilfe von *Vulture*-Plugins an eine Vielzahl von Management-Plattformen anbindbar und scripting-fähig.

Die Anbindung von *Vulture*-Agenten an eine Management-Plattform erfolgt über ein plattformspezifisches *Vulture*-Plugin. Ein *Vulture*-Plugin agiert als in die Management-Station integriertes Frontend zum Agenten. Soll *Vulture* in eine bisher nicht unterstützte Management-Plattform integriert werden, ist lediglich ein neues Plugin zu implementieren.

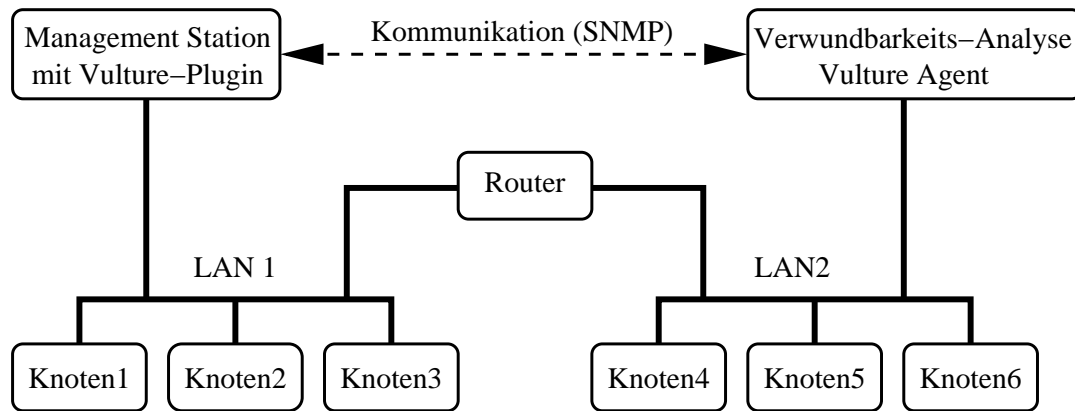


Abbildung 1: *Vulture*-Plugin und *Vulture*-Agent im Netzwerk

Während ein Plugin mehrere Agenten ansteuern kann, kann ein Agent mehrere Plugins bedienen. Dementsprechend können Agenten beliebig in der IT-Infrastruktur platziert werden und von verschiedenen Stellen und Administratoren genutzt werden.

Die Netzwerk-Management-Station kann von allen im Netz verteilten *Vulture*-Agenten die Version abfragen und ggf. eine aktuelle Liste aller angebotenen Verwundbarkeits-Tests inklusive der Hilfetexte und Standard-Parameter abrufen.

3.1 Sicherheit

Einige ältere Versionen von SNMP bieten nur unzureichende Sicherheitsmerkmale, so daß möglichst SNMPv3 verwendet werden sollte. Leider unterstützen noch nicht alle Netzwerk-Management-Stationen diese Version. Einen kurzen Überblick über die Risiken der verschiedenen SNMP-Versionen und welche Mindestmaßnahmen zu deren Minimierung empfehlenswert sind findet man unter [6].

Die fein-granulare Nutzer- und Rechteverwaltung von *Vulture* basiert auf dem VACM-Modell [7] und ist dadurch aus der Ferne wartbar. Es erlaubt die Verwaltung von Nutzern, Gruppen und MIB-Views (siehe Abschnitt 3.4.2) sowie die Zuweisung separater Rechte je nachdem, mit welcher SNMP-Version der Zugriff erfolgt.

Bei besonders sicherheitskritischen Tests sollte eine Authentifizierung und / oder Verschlüsselung der SNMP-Verbindung nach dem USM-Modell [8] von SNMPv3 [9] vorgeschrieben werden, bei weniger kritischen Tests mag auch eine einfache Verbindung nach SNMPv1 [10] genügen. Dies erlaubt eine eingeschränkte Nutzung auch von Netzwerk-Management-Stationen aus, die nur ältere SNMP-Versionen unterstützen. Detaillierte Informationen über die Sicherheitsfeatures von SNMPv3 kann man bei William Stallings finden [11, 12].

Als ein mögliches Einsatzszenario kann man für die verschiedenen Administratoren der einzelnen Abteilungen einer Organisation jeweils eine eigene Nutzergruppe vorsehen, die jeweils nur das Recht hat, ihr eigenes Abteilungsnetz zu testen. Durch aufstellen von mehreren *Vulture*-

Agenten an verschiedenen Stellen des Gesamtnetzes können die jeweiligen Administratoren testen, ob Ihr lokales Netz vom jeweiligen Agenten-Standort aus angreifbar ist.

3.2 Test-Taxonomie

Um eine leichtere Auswahl der Verwundbarkeits-Tests zu ermöglichen kann man diese in Taxonomien klassifizieren. Es existieren bereits einige Arbeiten über die Klassifikation von Verwundbarkeits-Tests, die allerdings zu sehr unterschiedlichen Einteilungen kommen, da nach unterschiedlichen Kriterien wie etwa der verwendeten Technik [13], den ausgenutzten Sicherheitslücken [14] oder den erzielten Resultaten [15] klassifiziert worden ist.

Deshalb bietet *Vulture* die Möglichkeit, beliebige Taxonomien anzulegen und für jeden Verwundbarkeits-Test die entsprechende Position im jeweiligen Taxonomiebaum festzulegen. Dadurch ist es dem jeweiligen *Vulture*-Plugin möglich, vielfältige Selektionsmöglichkeiten anzubieten.

3.3 Rückmeldungen

Nachdem der jeweilige Verwundbarkeits-Test beendet ist, sind seine Ausgaben in einer SNMP-Tabelle mit Zusatzinformationen wie etwa Zeitstempeln von Beginn und Ende der Ausführung abrufbar. Insbesondere für Skripte ist die Möglichkeit gedacht, reguläre Ausdrücke auf diese Ausgaben anwenden zu können und die Ergebnisse in einer eigenen SNMP-Tabelle bereit zu stellen. Für die Kommunikation mit dem Agenten kann entweder eine SNMP-Bibliothek der jeweiligen Skriptsprache benutzt werden, wie sie etwa in Perl [16] zur Verfügung steht, oder aber die `snmpget`- und `snmpput`-Kommandos aus dem NET-SNMP Paket verwendet werden.

Außerdem wird eine IDMEF-Nachricht [17] im XML-Format erstellt. Dieses Datenformat dient vor allem der Kommunikation von Intrusion-Detection-Systemen untereinander bzw. mit Management-Stationen und ermöglicht so die Korrelation von Angriffen. Sie enthält Informationen über den Detektor, Zeitstempel, Angaben über Quelle und Ziel sowie Art und Technik der vom Test ausgelösten Schutzzielverletzung und, ob diese erfolgreich war. Geplant ist, diese Nachricht der Alarm-Korrelation zuzuführen. Es besteht so die Möglichkeit, vom IDS gemeldete Schutzzielverletzungen als harmlos einzustufen, wenn sie durch eine Verwundbarkeits-Analyse ausgelöst wurden (s. Abschnitt 4).

In einer weiteren SNMP-Tabelle wird die *Intrusion Detection Sensor Alert MIB* [18] realisiert, die ähnliche Daten wie die IDMEF-Nachricht bereitstellt, allerdings nicht als XML-Dokument, sondern als einzelne Werte auf die dadurch gezielt per SNMP zugegriffen werden kann.

Außerdem werden alle Zugriffe auf den Agenten mittels *Syslog* protokolliert, um später nachvollziehen zu können, wer wann welchen Test gegen welches Ziel veranlaßt hat.

3.4 Implementierung und Konfiguration

Die Implementierung wurde im Rahmen einer Diplomarbeit [19] erstellt und wird frei verfügbar gemacht unter folgender URL: <http://ls6-www.cs.uni-dortmund.de/vulture>

3.4.1 *Vulture*-Plugin

Mittels *Vulture*-Plugins werden dem Administrator auf seiner gewohnten Management-Oberfläche *Vulture*-Agenten verfügbar gemacht. Eine Umgewöhnung entfällt und *Vulture* nutzt das Infrastruktur-Modell der Management-Station. Ziele für Verwundbarkeits-Analysen können u.a. im graphischen Infrastruktur-Modell ausgewählt werden (s. Abb. 2). Die Ergebnisse der Verwundbarkeits-Analyse erscheinen ebenfalls auf der Management-Oberfläche.

Bei der Auswahl der zu fahrenden Verwundbarkeits-Tests wird der Administrator durch frei definierbare Taxonomien unterstützt, z.B. wenn protokoll- oder betriebssystemspezifische Tests gewünscht sind. Andere Taxonomien ordnen Verwundbarkeits-Tests nach Seiteneffekten oder Angriffs-Sorten ein, z.B. Denial of Service (s. *Select Attacks* in Abb. 2).

Die Verwundbarkeits-Tests sind bereits vorparametrisiert, die Parameter sind aber einstellbar (s. *Firewalk Parameters* in Abb. 2). Informationen über die aktuell verfügbaren Verwundbarkeits-Tests und deren Parameter bezieht das Plugin vom Agenten.

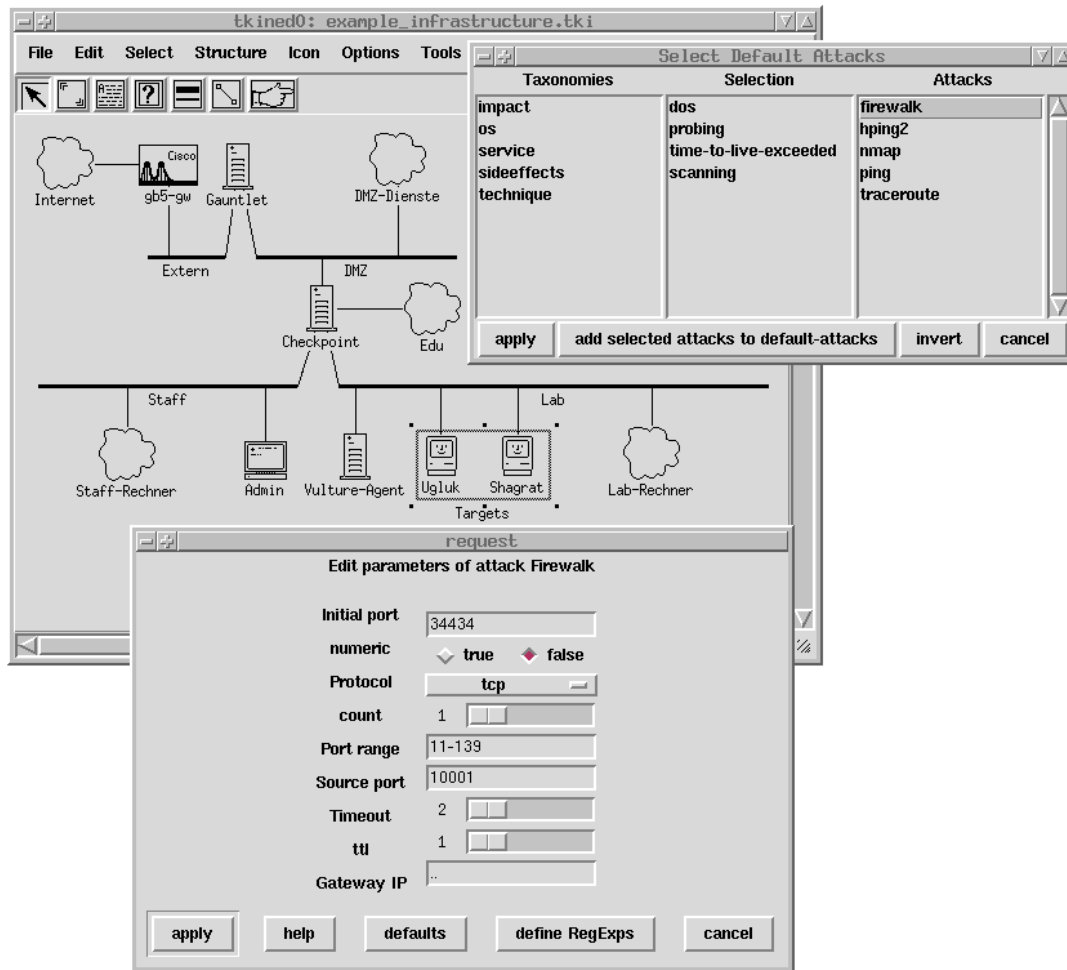
Implementiert wurde bisher ein Plugin für die frei verfügbare OpenSource Management-Plattform *Scotty/TKined*.

3.4.2 *Vulture*-Agent

Vulture-Agenten stellen den *Vulture*-Plugins die Verwundbarkeits-Tests zur Verfügung. Hierfür setzen die Agenten verschiedene weit verbreitete Standards ein. Die Kommunikation mit den Plugins erfolgt mittels SNMP und den entsprechenden Sicherheitsfeatures (SNMPv1, v2c, v3 USM). Daher sind die Agenten auch über die Kommandozeile fernsteuerbar und damit scripting-fähig.

Die Verwundbarkeits-Tests werden vom Agenten in einer erweiterbaren MIB [20] verwaltet. Die MIB enthält die Informationen zu den einzubindenden Verwundbarkeits-Testwerkzeugen, also deren Parameter- und Ausgabeformate, ihre Zuordnung zu Taxonomien, entsprechende Hilfstexte, etc. (s. Abb. 3). Der Agent startet jeden angeforderten Verwundbarkeits-Test gemäß MIB als eigenen Thread und verarbeitet dessen Ausgaben. Über einen MIB-Abgleich erhalten die Plugins vom Agenten die Informationen über die verfügbaren Verwundbarkeits-Tests und erzeugen daraus die grafischen Oberflächenelemente.

Die Implementierung der *Vulture*-Agenten basiert auf den Bibliotheken des frei verfügbaren OpenSource-Pakets *NET-SNMP* [21], das eine robuste SNMP-Engine bietet und alle SNMP-Versionen unterstützt.

Abbildung 2: Scotty/TKined mit *Vulture*-Plugin

```

initialPort OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "title: Initial port
         option: ' -I '
         GUItype: entry 10
         rem: initial destination port for TTL ramping."
    DEFVAL { 34434 }
    ::= { firewalk 1 }
  
```

Abbildung 3: Definition eines Parameters des firewalk-Tests in der *Vulture*-MIB

Die vorhandene Konfigurationssyntax wurde um zwei Schlüsselworte erweitert. Bei der Rechtevergabe ist es dadurch mittels `target` möglich, eine Menge von Netzknoten festzulegen,

die als potentielle Ziele der Verwundbarkeits-Tests erlaubt sind. Danach kann die definierte Zielmenge mit `allow` einer Nutzergruppe zugewiesen werden.

Abb. 4 zeigt eine Beispielkonfigurationsdatei. Die ersten beiden Zeilen liefern nötige Angaben zur Generierung von IDMEF-Nachrichten. MIB-Views definieren jeweils einen oder mehrere Teilbäume der MIB, auf die von Nutzern zugegriffen werden darf, wenn die entsprechenden Rechte vergeben sind.

Darunter wird eine Menge von erlaubten Zielen mit dem Namen `testtargets` angelegt sowie eine Nutzergruppe `tester` die nur den Nutzer `kommnick` enthält. Dieser kann die Verwundbarkeits-Tests auf die erlaubten Ziele starten, wenn er authentifiziert (`auth`) auf diesen Teilbaum der MIB (`attackview`) zugreift. Außerdem darf er auch ohne Authentifizierung auf den Teilbaum `harmlessview` zugreifen, da die darin enthaltenen Tests vom Administrator als harmlos eingestuft wurden.

Die Nutzergruppe `admin` enthält den Nutzer `flegel`, der auf den VACM-Teilbaum `adminview` nur verschlüsselt und authentifiziert (`priv`) zugreifen darf. Die auskommentierte Zeile würde den Nutzer `flegel` mit den entsprechenden Passwörtern erzeugen und verschlüsselt in einer weiteren Konfigurationsdatei anlegen. Das ist nur beim erstmaligen Start des *Vulture*-Agenten notwendig. Danach ist es per Fernwartung möglich, diese Passwörter zu ändern, sowie neue Nutzer und Gruppen zu erzeugen und zu verwalten.

Die Gruppe `update` erlaubt es, auf den Bereich der MIB zuzugreifen, der Informationen über die angebotenen Verwundbarkeits-Tests, deren Parameter und Hilfetexte enthält. Zuletzt werden noch Netzknoten definiert, an die der Status eines Tests als Benachrichtigung (SNMP-Trap) geschickt werden soll.

3.5 Ablauf einer Sitzung

Nachdem einmalig dem *Vulture*-Plugin die *Vulture*-Agenten mit den nötigen Parametern bekannt gemacht wurden, stehen diese zur Auswahl in einer Liste bereit. Daraus kann dann jeweils die Kombination von Agenten ausgewählt werden, welche die gewünschten Tests ausführen sollen.

Danach selektiert man die gewünschten Verwundbarkeits-Tests, was einem durch die Taxonomie-Auswahl erleichtert wird (Select Default Attacks-Fenster). Durch einen Doppelklick auf einen der Tests ist eine Anpassung seiner Parameter möglich.

Nachdem man dann die zu testenden Ziele selektiert hat (Ugluk und Shagrat in Abb. 2), kann man über den Menüpunkt `Start Selected Tests` die Tests starten.

Abb. 5 zeigt das Fenster mit der Zusammenfassung der Ergebnisse. Nach einer fortlaufenden Numerierung erscheint der Name des getesteten Ziels, des Agenten und des Tests sowie die Start- und Endzeit des Tests. Schließlich wird noch der Rückgabewert, den der jeweilige Test nach seiner Beendigung liefert (i.d.R. 0 bei Erfolg) und der Name des aufgerufenen Tests samt Pfad und Parametern ausgegeben. Die Meldung `pending` anstelle der Endzeit deutet an,

```
AnalyzerIdent 1234567890
AnalyzerLocation Room 12 Building 4

view allview      included .1.3.6.1
view adminview    included .1.3.6.1.6.3.16.1
view updateview   included .1.3.6.1.4.1.2176.255.1
view attackview   included .1.3.6.1.4.1.2176.255
view harmlessview included .1.3.6.1.4.1.2176.255
view harmlessview excluded .1.3.6.1.4.1.2176.255.4
view harmlessview included .1.3.6.1.4.1.2176.255.4.2
view harmlessview included .1.3.6.1.4.1.2176.255.4.3

target testtargets 193.98.235.0/24
target testtargets 193.102.98.0/24
target testtargets firewall.udo.edu

group tester usm kommnick
allow tester testtargets
access tester "" usm auth exact attackview attackview none
access tester "" usm noauth exact harmlessview harmlessview none

group admin usm flegel
access admin "" usm priv exact adminview adminview adminview
#createUser flegel MD5 flegel-auth-password DES flegel-priv-password

com2sec updateuser 0.0.0.0/0 public
group update v1 updateuser
group update v2c updateuser
group update usm updateuser
access update "" any noauth exact updateview none none

trap2sink scotty.udo.edu      public 162
trap2sink traplogger.udo.edu  public 162
```

Abbildung 4: Beispielkonfiguration des *Vulture*-Agenten

daß dieser Test noch nicht beendet ist. Sollte bereits der Start eines Tests fehlschlagen, etwa wegen falscher Parameter für die Kommunikation mit dem *Vulture*-Agenten, wird anstatt der Zeitstempel die Meldung `failed` ausgegeben.

Es ist möglich, eine oder mehrere Zeilen zu selektieren und sich mittels der in Abb. 5 gezeigten Buttons die Ergebnisse anzeigen zu lassen, noch laufende Tests abubrechen oder etwa die Ergebnisse abzuspeichern.

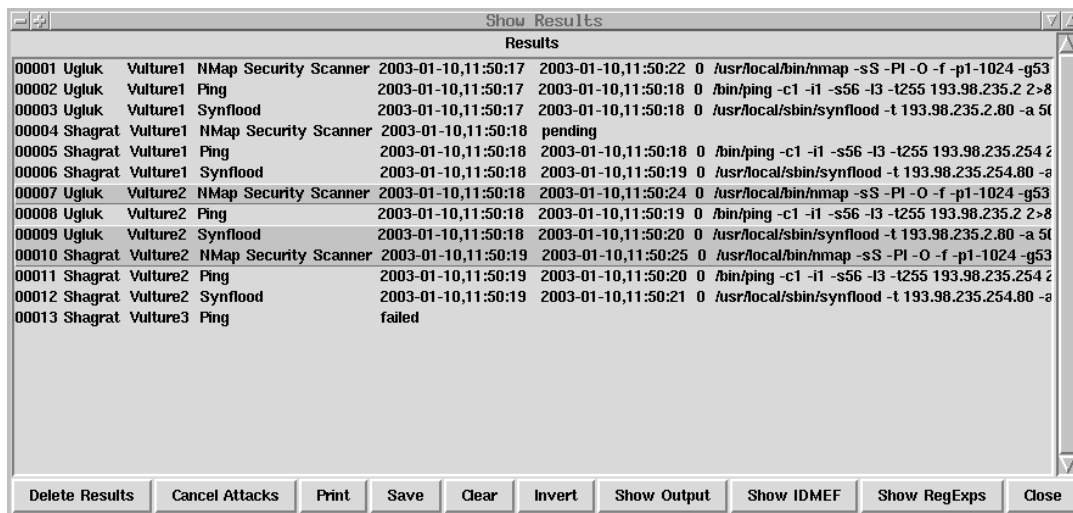


Abbildung 5: Ergebnisfenster

4 Ausblick

Vulture bietet heute die notwendige Basis-Technologie für die Integration von Verwundbarkeits-Analyse und Netzwerk-Management. Es sind vielfältige Erweiterungen denkbar und einige davon geplant.

Zunächst wären graphische Nutzeroberflächen für die Verwaltung von Verwundbarkeits-Tests, Nutzern und Rechten hilfreich.

Zur Unterstützung der Administratoren wäre eine Analyse-Ergebnis-Datenbank wünschenswert, in der Analyse-Ergebnisse automatisch konsolidiert werden. Eine umfangreichere Reporting-Funktionalität wäre auf Basis dieser Datenbank umsetzbar.

Die in den Beispiel-Szenarien vorgeschlagene automatisierte Unterstützung von Querbeziehungen zwischen Verwundbarkeits-Analyse und Netzwerk-Management ist auf der Basis von *Vulture*-Plugins umsetzbar (vgl. Abschnitt 2.1).

Die Integration von Verwundbarkeits-Analyse und Intrusion-Detection sind vorbereitet. Hier sind ebenfalls Querbezüge nutzbar. Beispielsweise könnte ein *Vulture*-Agent das IDS über eine laufende Verwundbarkeits-Analyse informieren. Das IDS kann die auftretenden Alarmmeldungen entsprechend sinnvoll einstufen.

Literaturverzeichnis

- [1] Fyodor. Exploit World, 2003. <http://www.insecure.org/sploits.html>.
- [2] Renaud Deraison. Nessus 1.3.0, January 2003. <http://www.nessus.org/intro.html>.
- [3] HP OpenView. <http://www.openview.hp.com/>.
- [4] IBM Tivoli. <http://www.tivoli.com/inside/companyinfo/background/>.
- [5] Jürgen Schönwälder. Scotty network management software package. <http://www.ibr.cs.tu-bs.de/projects/scotty>.
- [6] J. L. Camacho. SNMP Security Enhancement, September 2001. http://www.sans.org/rr/netdevices/SNMP_sec.php.
- [7] B. Wijnen, R. Presuhn, and K. McCloghrie. RFC 2275: View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), January 1998.
- [8] U. Blumenthal and B. Wijnen. RFC 2274: User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), January 1998.
- [9] D. Levi, P. Meyer, and B. Stewart. RFC 2273: SNMPv3 Applications, January 1998.
- [10] J. D. Case, M. Fedor, M. L. Schoffstall, and C. Davin. RFC 1157: Simple Network Management Protocol (SNMP), May 1990.
- [11] W. Stallings. Snmv3: A Security Enhancement for SNMP. *IEEE Communications Surveys*, 1(1), Fourth Quarter 1998.
- [12] W. Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley Longman Publishing Co., Inc., third edition, 1998.
- [13] P. G. Neumann and D. B. Parker. A summary of computer misuse techniques. In *Proceedings of the 12th National Computer Security Conference*, pages 396–407, 1989.
- [14] T. Aslam, I. Krsul, and E. H. Spafford. Use of a Taxonomy of Security Faults. In *Proc. 19th NIST-NCSC National Information Systems Security Conference*, pages 551–560, 1996.
- [15] U. Lindqvist and E. Jonsson. How to Systematically Classify Computer Security Intrusions. In *Proceedings of the 1997 IEEE Symposium on Security & Privacy*, pages 154–163, 1997.

- [16] Larry Wall. Perl 5. <http://www.perl.com/pub/q/documentation>.
- [17] D. Curry and H. Debar. Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition, November 2002. Internet Draft `draft-ietf-idwg-idmef-xml-09.txt`.
- [18] G. Mansfield and D. Gupta. Intrusion Detection Sensor Alert MIB, November 2000. Internet-Draft `draft-glenn-id-sensor-alert-mib-01.txt`.
- [19] M. Kommnick. Integration von Werkzeugen für die Verwundbarkeits-Analyse in Netzwerk-Management-Umgebungen. Master's thesis, Universität Dortmund, FB4-LS6-ISSI, D-44221 Dortmund, August 2002.
- [20] K. McCloghrie and M. T. Rose. RFC 1156: Management Information Base for network management of TCP/IP-based internets, May 1990.
- [21] NET-SNMP 5.0.6, October 2002. <http://www.net-snmp.org>.