Security Analysis Tool (SAT) 2.0

Visualisierung von Rechtestrukturen

SAT – Viewer – MMC-SnapIn

Gerald Zarda
a member of the SAT-team at FIM

FIM – Institut für Informationsverarbeitung und Mikroprozessortechnik
Johannes Kepler Universität Linz, A-4040 Linz, Austria
Email: sat@fim.uni-linz.ac.at, Web: http://www.fim.uni-linz.ac.at/sat

Dank

Dieses Projekt wird von
Microsoft Research Cambridge UK
(http://www.research.microsoft.com/labs/cam.asp)
unterstützt.

Inhalt

- I. Das SAT Projekt
- II. Security Analysis Tool (SAT) 2.0

I. Das SAT Projekt

- Motivation
- Geschichte
- SAT 1.0 – Architektur
- Ziele

SAT Projekt – Motivation [1/2]

- Hoher Stellenwert der Netzwerksicherheit
- Mehr als Firewalls, etc.
- „Innere Sicherheit"
- Besonders wichtig beim Einsatz von Technologien wie dem ADS
- Verwaltung von Benutzerberechtigungen stellt eine komplexe Aufgabenstellung für Administratoren dar

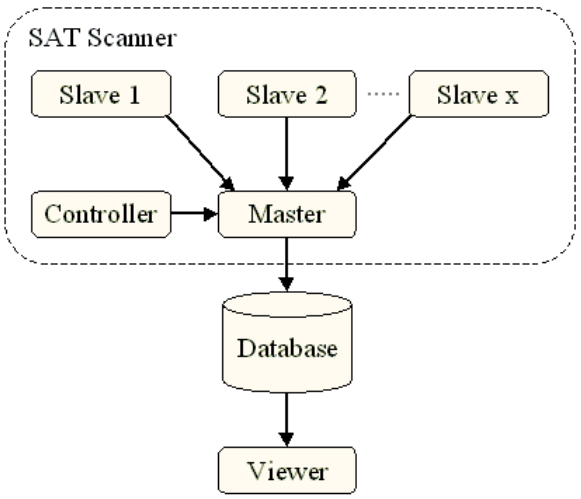
SAT Projekt – Motivation [2/2]

- Objekt-zentrierte Sicht
  - Berechtigungen können mit Windows Standard-Werkzeugen nur für einzelne Objekte oder Gruppen von Objekten vergeben und dargestellt werden
- Benutzer-zentrierte Sicht
  - Fragestellung: „Wo hat Benutzer X Rechte?"
  - Ergebnis: Auflistung aller Objekte, auf die ein bestimmter Benutzer oder eine Gruppe Rechte hat (wird von Windows NT bzw. Windows 2000/XP nicht unterstützt)
  - Lösung: Security Analysis Tool (SAT)

SAT Projekt – Geschichte

- SAT 1.0
  - Grund: Eingeschränkte Möglichkeiten bei der Verwaltung von Benutzerberechtigungen unter Windows NT
  - Entwickler: Rudolf Hörmanseder, Kurt Hanner
  - für NT 4.0 Server
  - zur Analyse von Benutzer-Rechten im NTFS
  - Speicherung in Access-DB
  - Visualisierung mittels MFC-Applikation
  - Version 1.0B wurde im Nov. 1999 fertiggestellt

SAT Projekt – Architektur SAT 1.0



SAT Projekt – Ziele

- Security Analysis Tool 2.0
  - „Wo hat User X Rechte?"
  - Windows 2000 Systeme
  - Analyse von
    - Active Directory
    - File System
    - Registry
  - Besseres/schnelleres Datenbanksystem
  - Visualisierung mittels MMC Snap-In

II. Security Analysis Tool 2.0

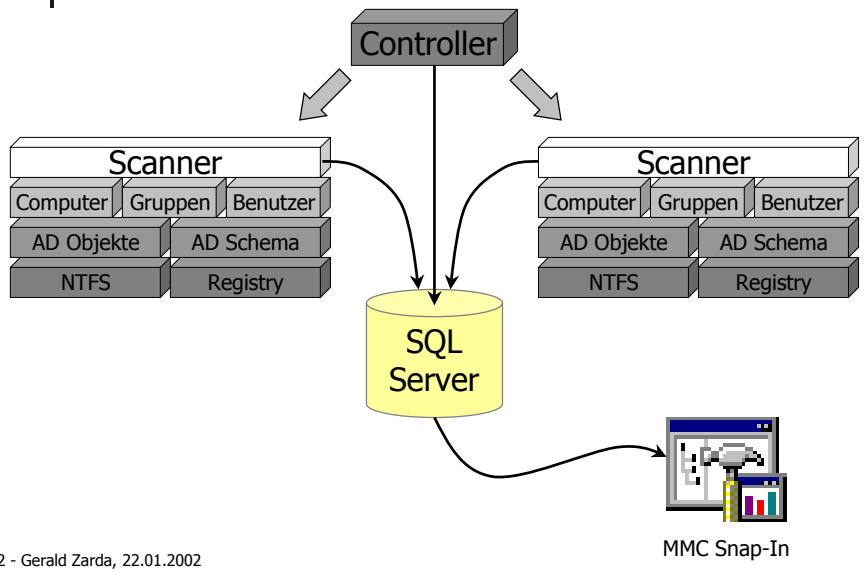
- SAT2 – Team
- Architektur
- Datenbank
- Controller/Master
- Datensammler
- Visualisierung
- Ausblick

SAT2 – Team

- **Hörmanseder**: techn. Projektleitung
- **Achleitner**: NTFS, Registry
- **Helml**: Active Directory, Datenbank
- **Zarda**: Computer-, Gruppen- u. Benutzerdaten, Mitgliedschaften, Visualisierung
- **Schmitzberger**: Controller/Master

- Finanzielle Unterstützung durch
  MSR – Microsoft Research, Cambridge (UK)

SAT2 – Architektur



SAT2 – Datenbank

- Microsoft SQL Server 2000
- Zugriff über ActiveX Data Objects (ADO)
- DB Funktionen implementiert in C++
- Scanner schreiben direkt in die DB
- Caching um Performance zu steigern
  - z.B.: ACLs im Active Directory (Basis-Installation)
    - ~ 2300 Objekte gescannt
    - ~ 60 verschiedene ACLs

SAT2 – Controller/Master

- Anlegen der DB und Tabellen
- Schreiben von Basis-Daten
- Schreiben/Lesen Registry-Informationen
  - Scanner-Konfiguration (Was?, Komprimierung, ...)
  - Cache-Limit
  - DB-Server Name
  - ...
- Aktivieren und Starten der Scanner

SAT2 – Datensammler

- CGU Scanner
- ADS Scanner
- NTFS Scanner
- REG Scanner

Datensammler – CGU Scanner [1/2]

- Sammelt/speichert Informationen über:
  - Computer
    - SID (Domain od. Local), NetBIOS Name, DNS Namen
  - Gruppen
    - SID, Name, Kommentar, Computer ID
  - Benutzer
    - SID, (Account) Name, Kommentar, Computer ID
  - Mitgliedschaften
    - SID, MemberOfSID

Datensammler – CGU Scanner [2/2]

- Implementierung
  - Entwicklungsumgebung:
    - MS Visual Studio 6.0, C/C++
  - Anmerkungen:
    - Std. Befehle zum Auslesen der Daten
    - Wenig Informationen über Computer
    - Unterschiedliches Auslesen der SIDs bei DC u. WS
    - Probleme beim Auslesen von Gruppen- u. Benutzerdaten
    - Speichern der (indirekten) Mitgliedschaften

Datensammler – ADS Scanner

- Sammelt/speichert Informationen über:
  - AD Objekte und/oder AD Schema
  - SID-History
- Caching von ACLs
- Komprimierungsstufen
  - Stufe 1: Zusammenfassen von gleichen Objekten
  - Stufe 2: nur Ausnahmen (Abweichung vom Regelfall)
  - Stufe 3: Kombination der Stufen 1 + 2

Datensammler – FS/REG Scanner

- Sammelt/speichert Informationen über:
  - NTFS (Directories, Files, MountPoints, ...)
  - Registry
- Caching von ACLs
- Komprimierungsstufen
  - Stufe 1: Zusammenfassen gleicher Objekte
  - Stufe 2: Zusammenfassen von Objekten eines Verz.
  - Optional: Nur Ausnahmen (Brüche), Zusammenfassen von Dokumenten gleicher Applikationen

SAT2 – Visualisierung

- Motivation
- Quickguide to MMC
- SAT2 V1.00 BETA – MMC Snap-In

Visualisierung – Motivation

- Gesammelte Informationen in übersichtlicher Form u. aussagekräftig darstellen
- Warum MMC ?
  - Standard für administrative Aufgaben
  - „Gewohnte Arbeitsbedingungen" für den Administrator
  - Verwendung vorhandener Snap-Ins und SAT2 in einer Applikation
  - Master wird als MMC Snap-In implementiert

Visualisierung – Quickguide to MMC

- Einleitung
- Konsolen und Snap-Ins
- Das Snap-In Basis-Framework
- Registrieren des Snap-Ins
- „About" – Information
- Namespaces
- GUI-Komponenten
- Systemvoraussetzungen

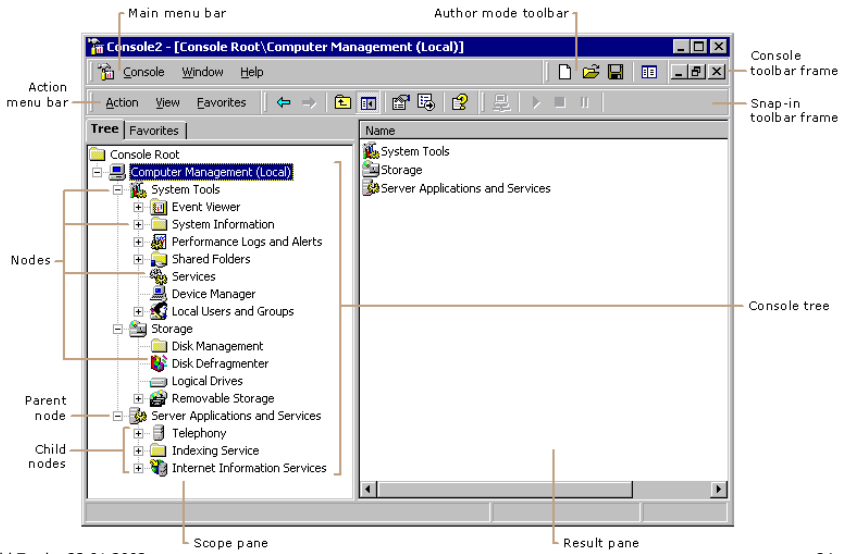
MMC – Einleitung

[1/2]

- Computer-/Netzwerk-Verwaltung stark geändert
- Schwierigkeiten für Administratoren:
  - Administrations-Tools u.U. schwer zu finden
  - Unterschiedliche Benutzeroberflächen
  - Nicht an Wissen oder Fähigkeiten d. Administrators anpassbar
- Lösung: Microsoft Management Console (MMC)
  - Entwickler: Tony Romano, Wayne Scott
  - 1. Prototyp im Juni 1996
- „MMC COM Programmer´s Guide" (Auszug)

MMC – Einleitung

[2/2]



MMC – Konsolen und Snap-Ins

- Microsoft Management Console (MMC)
  - Erweiterbares UI für Verwaltungs-Applikationen
  - Öffnen, Speichern od. Erzeugen von Administrations-Tools (MMC-Konsolen)
  - MMC selbst bietet keine administrative Funktionalität
  - Host für administrative Komponenten (Snap-Ins)
  - Multiple Document Interface (MDI)
  - Jedes Fenster ist eine eigene „View" und kann ein oder mehrere Snap-Ins enthalten
  - Stand-Alone od. Extension Snap-In

MMC – Das Snap-In Basis-Framework

- Funktionalität wird durch Snap-In bestimmt
- Snap-In = „COM in-process server DLL"
  - COM-Interface bildet Schnittstelle zwischen MMC und Snap-In
  - MMC benötigt kein „Wissen" über Snap-In
- Grundgerüst:
  - MMC COM Interfaces
    - IComponentData und IComponent
  - Standard COM Interfaces
    - IDataObject und IClassFactory

MMC – Registrieren des Snap-Ins

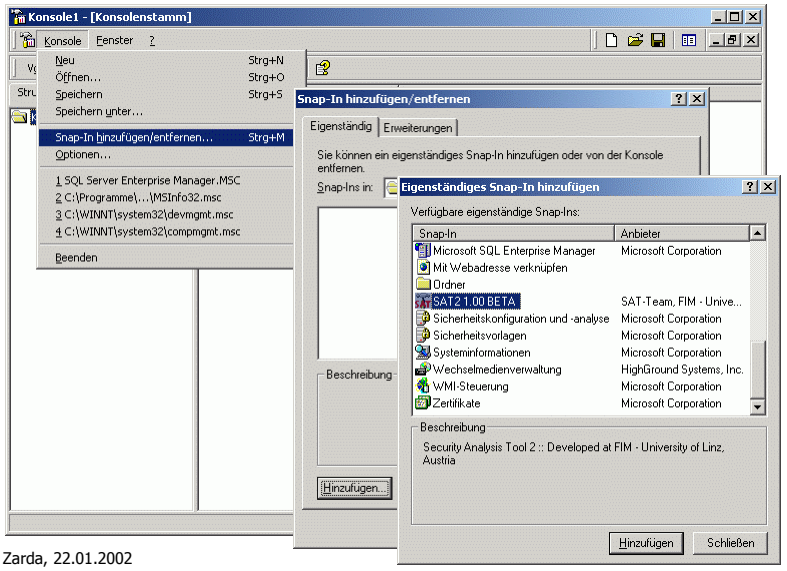
- COM in-process server DLLs müssen sich „selbst" registrieren
  - Spezieller Bereich der System-Registry:
    HKEY_LOCAL_MACHINE\Software\Microsoft\MMC\SnapIns
  - Snap-In Registry-Key

    HKEY_LOCAL_MACHINE\Software\Microsoft\MMC\SnapIns\ {*snapinCLSID*}
    About REG_SZ "{*SnapinAboutCLSID*}"
    NameString REG_SZ "*SnapinDisplayName*"
    StandAlone
    NodeTypes
    {*nodetypeGUID*}
  - Implementierung folgender Funktionen:
    - DllRegisterServer, DllUnregisterServer
    - RegisterSnapIn, UnregisterSnapIn

MMC – „About" - Information

- Ermöglicht Anzeige von
  - Snap-In Name
  - Provider
  - Versionsnummer
  - Beschreibung
- Nur sichtbar im „Add/Remove Snap-In" Dialog
- Daten werden in der Registry gespeichert
- Notwendig um „Static Node"-Image zu ändern
- Implementierung des ISnapinAbout Interfaces

Snap-In hinzufügen



MMC – Namespaces

- Bezeichnung für ScopePane u. ResultPane
- ScopePane
  - „Linke Seite" des Snap-In od. „Console Tree"
  - Wurzel: Static Node
  - Söhne: ScopeItems (Ordner bzw. Container)
  - Interface: IComponentData
- ResultPane
  - „Rechte Seite" des Snap-In
  - Anzeige von ScopeItems u. ResultItems
  - Verschiedene „Views" möglich (Taskpad, OCX, Web)
  - Interface: IComponent

MMC – GUI-Komponenten

- Interfaces für folgende GUI Elemente:
  - Context menus
  - Toolbars and menus
  - Property sheets
  - Wizards
  - Taskpads

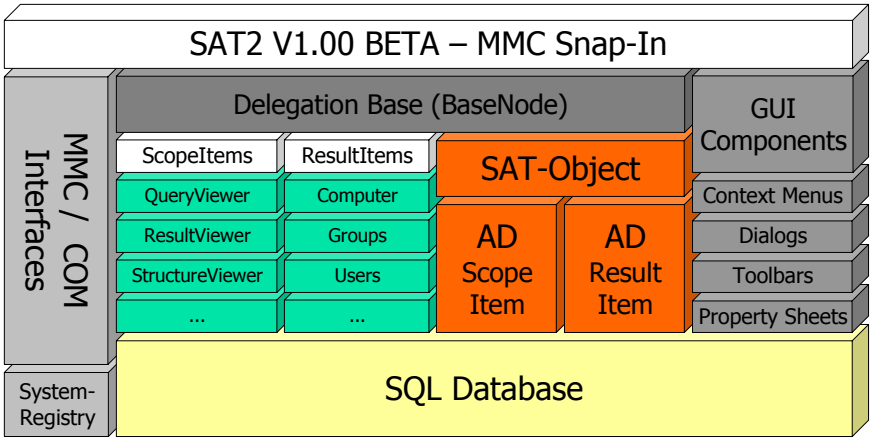
MMC – Systemvoraussetzungen

- Betriebssystem:
  - Microsoft Windows NT 4.0 (SP3)
  - Microsoft Windows 2000 (ab Beta 3)
  - Microsoft Windows 95 oder Windows 98
- Entwicklungsumgebungen:
  - Microsoft VC++ 5.0/6.0
    - ATL/COM Snap-in Wizard
  - Microsoft VB 6.0
    - Snap-in Designer für VB
- MMC und .NET ?

Visualisierung – SAT2 MMC Snap-In

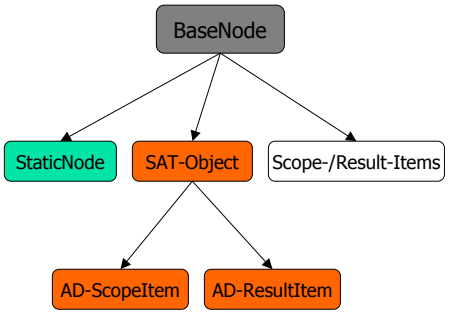
- Architektur
- GUI Komponenten
- Structure Viewer
- Query Viewer
- Result Viewer
- Erweiterbarkeit
- Probleme/Schwierigkeiten

SAT2 Snap-In – Architektur [1/2]

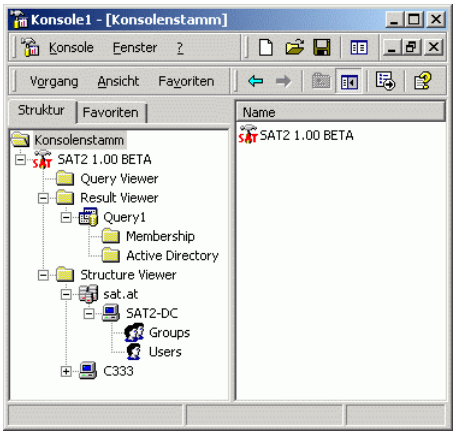


SAT2 Snap-In – Architektur [2/2]

- Objekt-Hierarchie (OOP)



- Objekt-Hierarchie (MMC)



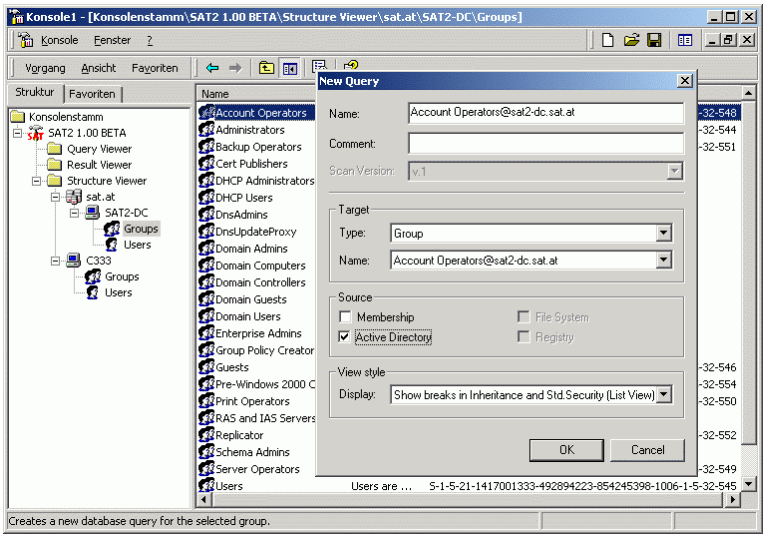
SAT2 Snap-In – GUI Komponenten

- Kontextmenüs u. Toolbar
  - Anlegen, Löschen u. Umbenennen von Abfragen
  - Aktualisieren der Anzeige
  - Aufrufen der Eigenschaftsseiten
  - Ausführen, Anhalten u. Abbrechen von Abfragen
- Dialoge („New"-Dialog)
  - Anlegen neuer Abfragen
  - Aufruf aus Query Viewer u. Structure Viewer
- Eigenschaftsseiten
  - Abfragen (Ändern d. Einstellungen)
  - Objekte im Result Viewer (Detailanzeige der Rechte)

SAT2 Snap-In – Structure Viewer

- Repräsentation des gescannten Netzwerks
  - Domains
    - Computer
      - Gruppen
      - Benutzer
- Möglichkeit neue Abfrage durch Auswahl des gewünschten „Ziel-Objekts" zu erzeugen (Kontext-Menü)

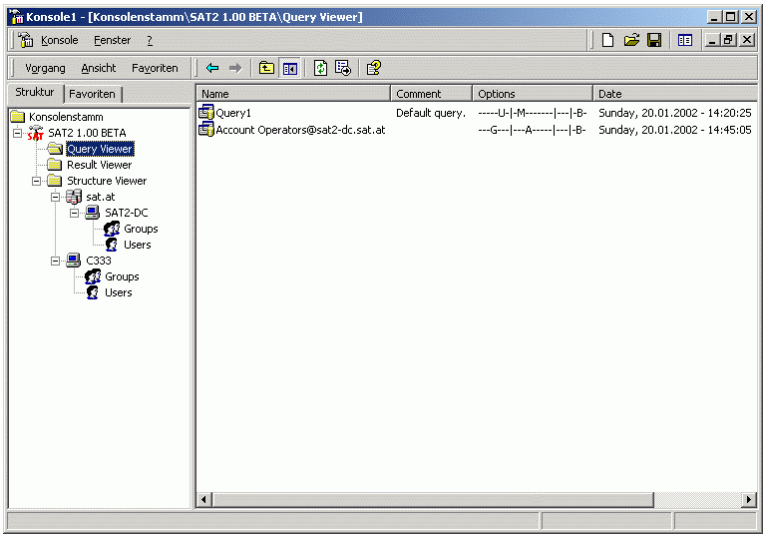
Structure Viewer



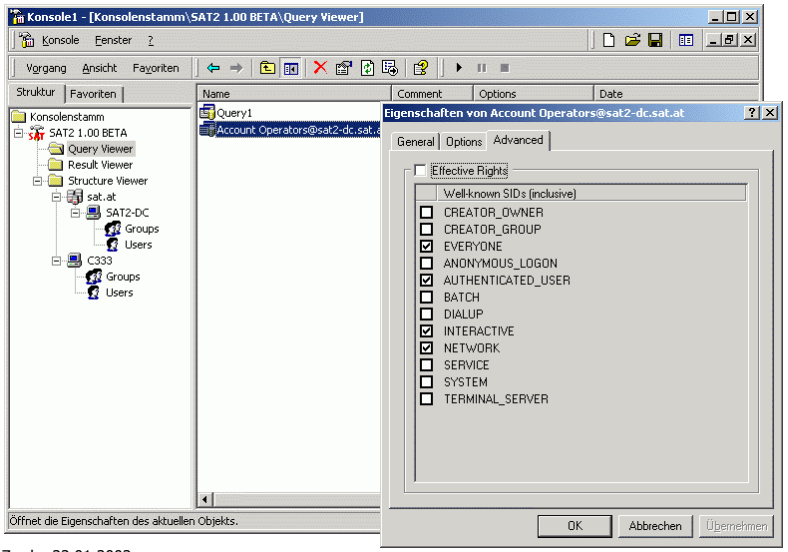
SAT2 Snap-In – Query Viewer

- Übersicht über Abfragen
- Mögliche Einstellungen für eine Abfrage
  - Name, Kommentar und Scanversion
  - Auswahl des „Ziels" (Wer?)
  - Auswahl der „Quelle" (Wo?)
  - Darstellungsarten
    - Listendarstellung
      - Lineare Liste
    - Baumdarstellung
      - Rekursiv oder „Aufbau bei Bedarf"

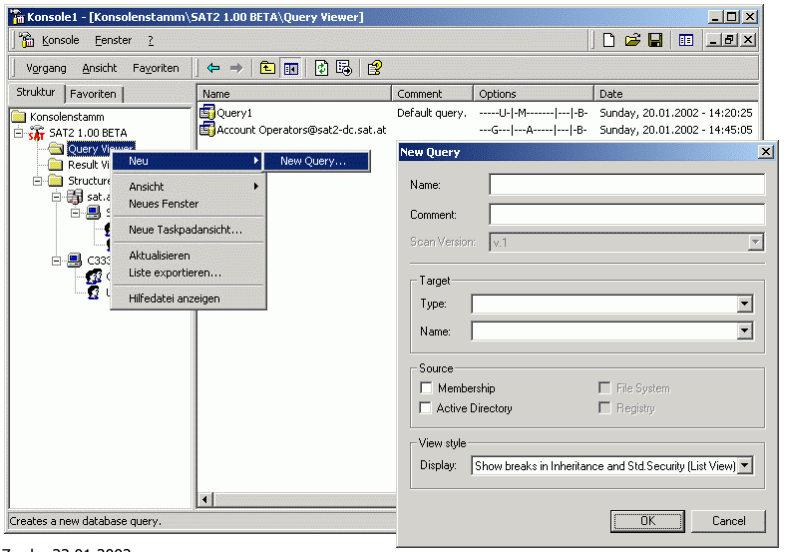
Query Viewer



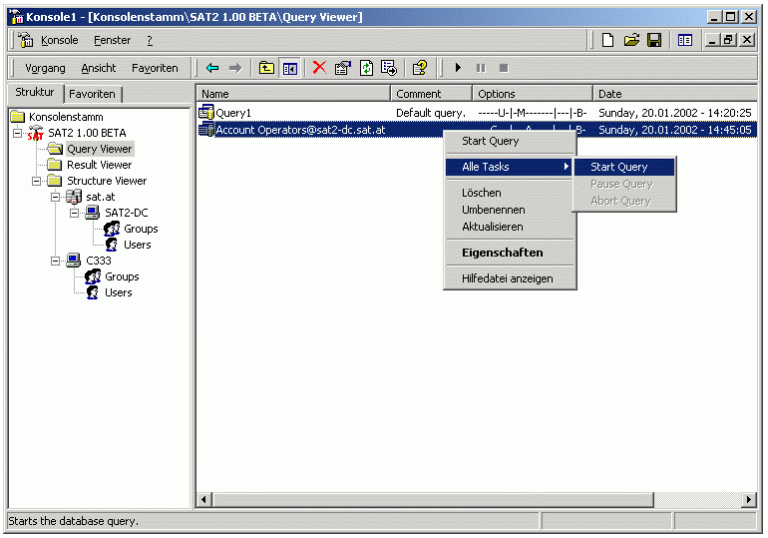
Query Viewer – Property Sheets



Query Viewer – New Dialog



Query Viewer – Start Query



SAT2 Snap-In – Result Viewer

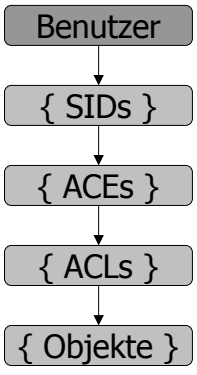
- Kernstück des Snap-In
- Analyse der gescannten Daten
- Darstellung der Daten
  - Ermittlung der Berechtigungen
  - Komprimierung der Rechte
  - Darstellungsarten
  - Einfärbung der Knoten

Ermittlung der Berechtigungen [1/3]

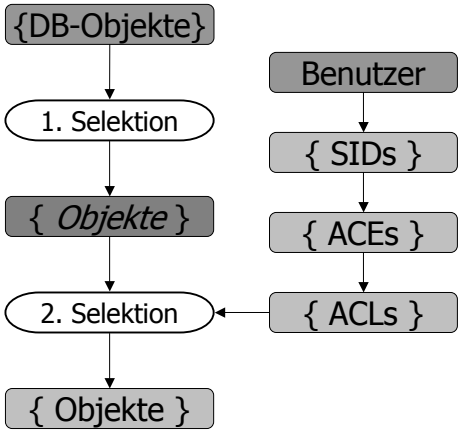
- Für jedes Objekt existiert eine ACL
- Jede ACL enthält einen oder mehrere ACEs
- Ein ACE enthält u.a. folgende Einträge:
  - SID: „Wer?"
  - Access Mask + ACE-Typ: „Was?"
  - Objekttyp: „Worauf?" (gilt nur für AD Objekte)
- Mittels SQL-Abfrage kann nun festgestellt werden, ob bzw. welche Rechte z.B. ein Benutzer auf ein bestimmtes Objekt hat

Ermittlung der Berechtigungen [2/3]

- Vorgehensweise:



- Verfeinerung:



Ermittlung der Berechtigungen [3/3]

- SQL-Abfrage:

```
SELECT o.obj_id FROM sat..objects AS o
WHERE  (o.breaksInheritance = '1' OR o.breaksStdSecurity = '1')
AND    acl_id IN
(
  SELECT a.acl_id FROM sat..ace AS a
  WHERE ( a.type = 'ADS_ACETYPE_ACCESS_ALLOWED' OR
          a.type = 'ADS_ACETYPE_ACCESS_ALLOWED_OBJECT' )
  AND   ( a.sid = 'szTargetSid'
          [ OR a.sid = 'szMemberOfSid' OR ... ] )
);
```


Komprimierung der Rechte [1/3]

- Einem AD Objekt können 19 verschiedene Standard-Rechte zugewiesen werden (Access Rights)
- gespeichert als Bitmaske (Access Mask)
- Ziel: übersichtliche Darstellung als String
- Schlechte Darstellung:

  ADS_RIGHT_DELETE, ADS_RIGHT_READ_CONTROL, ADS_RIGHT_WRITE_DAC, ADS_RIGHT_WRITE_OWNER, ADS_RIGHT_SYNCHRONIZE, ADS_RIGHT_ACCESS_SYSTEM_SECURITY, ADS_RIGHT_GENERIC_READ, ADS_RIGHT_GENERIC_WRITE, ADS_RIGHT_GENERIC_EXECUTE, ADS_RIGHT_GENERIC_ALL, ADS_RIGHT_DS_CREATE_CHILD, ADS_RIGHT_DS_DELETE_CHILD, ADS_RIGHT_ACTRL_DS_LIST, ADS_RIGHT_DS_SELF, ADS_RIGHT_DS_READ_PROP, ADS_RIGHT_DS_WRITE_PROP, ADS_RIGHT_DS_DELETE_TREE, ADS_RIGHT_DS_LIST_OBJECT, ADS_RIGHT_DS_CONTROL_ACCESS

- => Komprimierung

Komprimierung der Rechte [2/3]

- Abbildung der Access Mask auf einen String
- Form: [R-W-C-D] od. [r-w-c-d]
  - Buchstaben beliebig kombinierbar
  - Unterschied zw. Groß-/Kleinschreibung
    - Groß: Recht gilt für das „ganze" Objekt
    - Klein: Recht gilt nur für einen Teil d. Objekts (Attribut)
    - Wird anhand des ACE-Typs festgestellt
  - Legende:
    - R ... Read
    - W ... Write
    - C ... Create
    - D ... Delete

Komprimierung der Rechte [3/3]

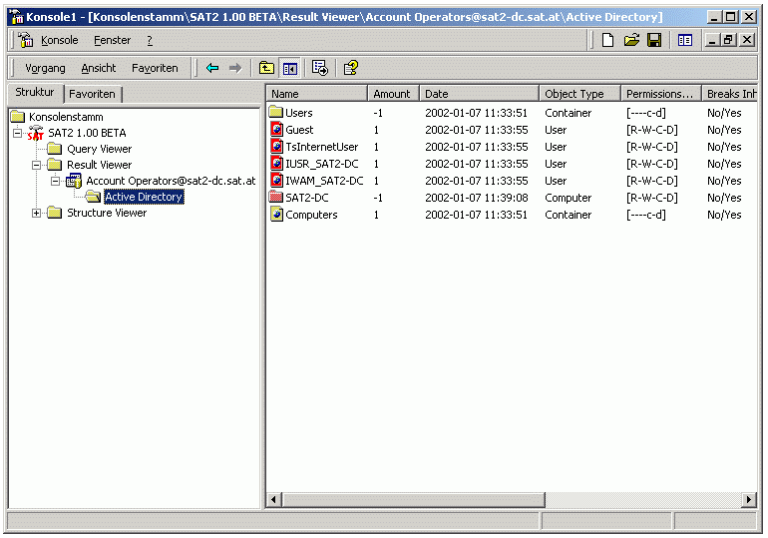
- Abbildung der Rechte

ADS_RIGHTS		ADS_ACETYPE_ACCESS_ALLOWED / ADS_ACETYPE_ACCESS_ALLOWED_OBJECT
ADS_RIGHT_DELETE	=>	D / d
ADS_RIGHT_READ_CONTROL	=>	R / r
ADS_RIGHT_WRITE_DAC	=>	W / w
ADS_RIGHT_WRITE_OWNER	=>	W / w
ADS_RIGHT_SYNCHRONIZE	=>	R / r
ADS_RIGHT_ACCESS_SYSTEM_SECURITY	=>	R / r + W / w
ADS_RIGHT_DS_CREATE_CHILD	=>	C / c
ADS_RIGHT_DS_DELETE_CHILD	=>	D / d
ADS_RIGHT_ACTRL_DS_LIST	=>	R / r
ADS_RIGHT_DS_SELF	=>	R / r + W / w
ADS_RIGHT_DS_READ_PROP	=>	R / r
ADS_RIGHT_DS_WRITE_PROP	=>	W / w
ADS_RIGHT_DS_DELETE_TREE	=>	D / d
ADS_RIGHT_DS_LIST_OBJECT	=>	R / r
ADS_RIGHT_DS_CONTROL_ACCESS	=>	R / r + W / w

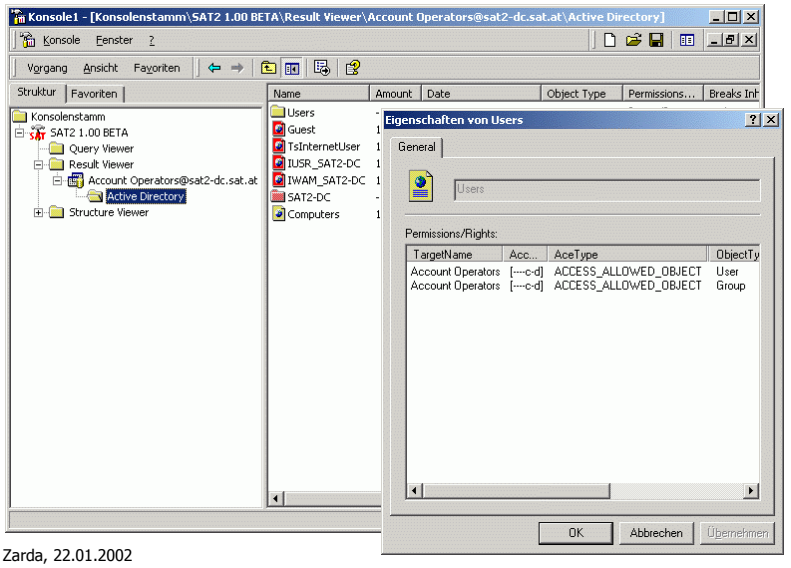
Darstellungsarten

- Darstellungsarten
  - Listendarstellung
    - Lineare Liste
    - Anzeige von Objekten
      - auf welchen das „Ziel" Rechte hat
      - Bruch in Standard Security und/oder Vererbung
    - Einfärbung:
      - Farben: Rot, Gelb, Grün
      - auf Objekt bezogen
      - Grün: Nur Leserechte
      - Gelb: Änderungsrechte
      - Rot: Full Control

Result Viewer – List View



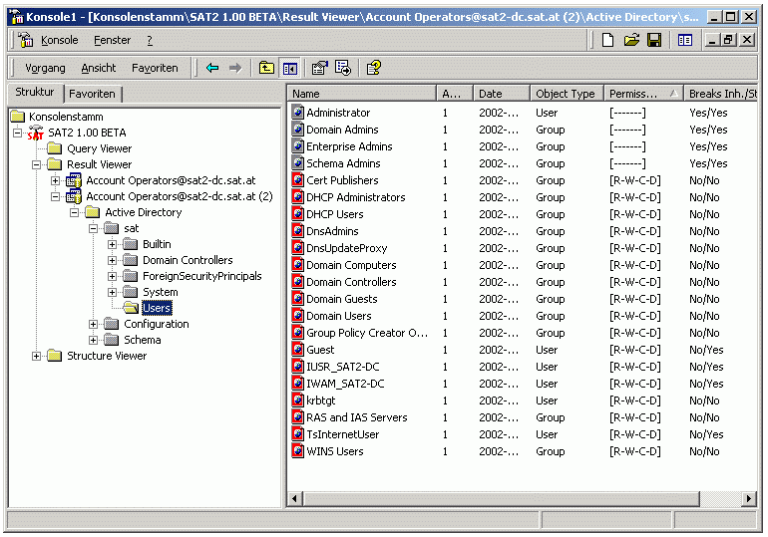
Result Viewer – Property Sheets



Darstellungsarten [2/3]

- Baumdarstellung (1)
  - Aufbau „On Demand"
  - Top Down – Analyse
  - Anzeige aller Objekte
  - Einfärbung:
    - Farben: Rot, Gelb, Grün u. Grau
    - auf Knoten/Blätter bezogen
    - Grau: Keine Rechte
    - Grün: Nur Leserechte
    - Gelb: Änderungsrechte
    - Rot: Full Control

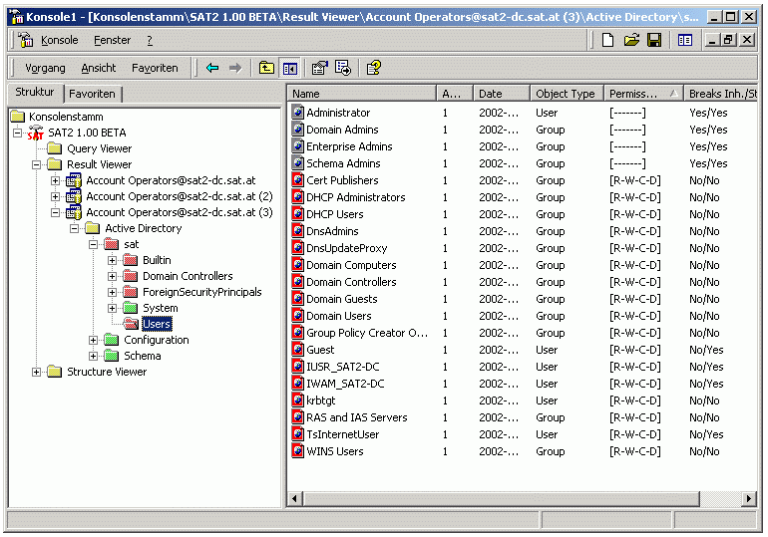
Result Viewer – Tree View



Darstellungsarten [3/3]

- Baumdarstellung (2)
  - Rekursiver Aufbau
  - Bottom Up – Analyse
  - Anzeige aller Objekte
  - Einfärbung:
    - Farben: Rot, Gelb, Grün u. Grau
    - Abhängig von der Farbe der Söhne
    - Grau: Keine Rechte (nur für Blätter)
    - Grün: Leserechte oder weniger
    - Gelb: Änderungsrechte
    - Rot: Full Control

Result Viewer – Extended TV



SAT2 Snap-In – Erweiterbarkeit

- Prototyp zur Weiterentwicklung
- Momentan nur AD Analyse
  - Daten standen als Erste zur Verfügung
  - Keine Erfahrungswerte
  - Komplexer als NTFS/REG Analyse
- Deshalb:
  - Basisklassen für NTFS/REG Analyse
  - 3 ausgewählte Darstellungsarten
  - Implementierung der GUI Komponenten

SAT2 Snap-In – Probleme

- Enormer Aufwand (Einarbeitung, Implementierung)
- Wenig Referenzen
- Platform SDK Code Beispiele fehlerhaft
- Beispiele:
  - MMC und MFC
  - ATL/COM Wizard
  - VB Snap-in Designer
  - „Property Sheets"

SAT2 – Ausblick

- Anzeigen von NTFS und REG Information
- Komprimierung der Anzeige
- Erweiterung der Abfragen
  - „Wo haben Benutzer mit gleichem Name Rechte?"
  - „Wo haben alle Mitglieder der Gruppe X Rechte?"
- Steigerung der Performance
- ALL-IN-ONE Snap-In
  - Steuerung
  - Datensammler
  - Visualisierung

Danke für Ihre Aufmerksamkeit!