HostLock – ein Quarantänesystem für Netzwerke

Adrian Wiedemann

Agenda

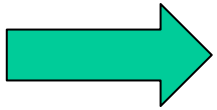
- Initiale Idee
- Anforderungen an System
- Modultypen
- Schematischer Aufbau
- Nachbehandlung von gesperrten Rechnern
- Implementierung an der Universität Karlsruhe
- Zusammenfassung

Initiale Idee

- Vorhandene IPS Sensoren im Netz
- Managementsoftware regelt die Angriffe an sich
- Keine Nachbehandlung von Vorfällen
- Händische Sperren am Router
- Händische Sperren von Benutzern
- On-demand Sperren

Anforderungen

- System muss dauerhaft infizierte Systeme erkennen können (kein On-Demand)
- Möglichst plattformunabhängig
- Möglichst sprachneutral
- Einfach erweiterbar

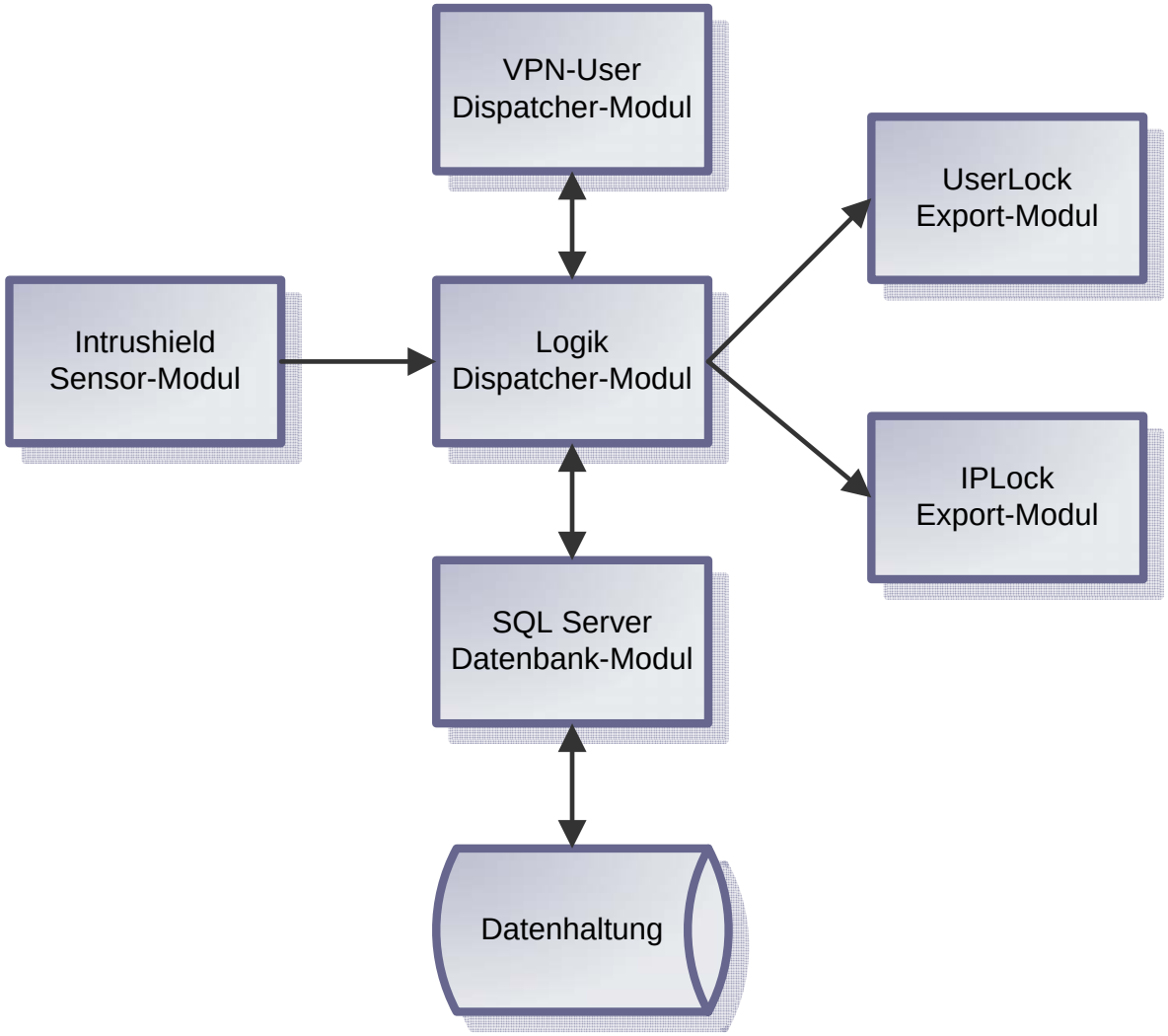


- Modularer Aufbau
- Standardisierte Schnittstellen
- Webservices, Kommunikation über HTTP/SOAP

Modultypen

- **Sensormodul**
  - Sammelt Daten von den Sensoren im Netz
- **Dispatchermodul**
  - Erhält Daten von Sensormodulen
  - Erhält den Arbeitsprozess für die unterschiedlichen Sensorendaten
- **Datenbankmodul**
  - Kapselt den Zugriff auf die Datenbank
- **Exportmodul**
  - Erzeugt je nach Anforderung unterschiedlichen Sperrmechanismus
    - Cisco Router ACLs
    - Firewall Regeln
- **Usermodul**
  - Erlaubt den Zugriff auf Daten und die Nachbearbeitung (Self-Service)

Schematischer Beispielaufbau



Nachbehandlung von gesperrten Rechnern

- Userportal für die Vorfallsbehandlung von gesperrten Rechnern
- Verwendung Service Oriented Architecture (SOA)
  - Möglichkeiten der Stellungnahme zum betroffenen System / Benutzer
  - Freigabe nach Begutachtung durch Netz-Betreiber
  - Einfache Möglichkeit der Definition weiterer Prozessabläufe
  - Einfache Erweiterbarkeit
- Sicherheitsaspekte
- Kommunikation nur zwischen geprüften Modulen erlauben
  - Gegenseitige Prüfung von Zertifikaten
  - PKI Infrastruktur notwendig

Implementierung an der Universität Karlsruhe

- Implementierung mittels .NET Webservices
- Datenhaltung noch auf SQL Server 2000
- Implementierung der Beispielskizze
- Momentan Testbetrieb – noch kein automatisiertes Sperren
- Usermodul noch nicht vorhanden

Zusammenfassung

- Ständiges Monitoring von Rechnern
- Möglichkeit der autonomen Reaktion
- System ist weitgehend sprach- und plattformneutral
- Modulbasiertes Design (SOA)
- Verwendung von Webservices für Inter-Modul Kommunikation
- Absicherung durch SSL-Zertifikate