Diplomarbeit - MAD Netzwerkmonitoring

Porcic Alin, Ranalter Daniel, Singh Manpreet, Stojanovic Marko

20. Juni 2014

Kapitel 1

Aufgabenstellung

Die Aufgabenstellung dieser Diplomarbeit beinhaltet das Schreiben einer Software welche für Administratoren im Heimnetzwerk oder kleineren Firmen ausgelegt ist. Sie soll dem Administrator verschiedenste Informationen über die in dem Netzwerk angeschlossenen Geräte, seien es Server oder Clients, Hosts u.a., bieten. Diese Informationen sollen über Hilfsmittel wie Ping und den Abfragen von auswählbaren Diensten erfolgen. Zu solchen zählen beispielsweise HTTP oder DHCP und DNS. Auch ein genereller Portscan soll mit dabei sein um eine ganze Reihe an Ports zu überprüfen, um so bei unwichtigeren Diensten grob überprüfen zu können ob zumindest der Port noch offen oder auch geschlossen ist. Im Endeffekt soll für den User unseres Programmes dann ersichtlich sein welche Adressen, MAC und IP, die angeschlossenen Hosts haben, sowie welche Ports von diesen offen sind. Das Programm wird für den User entweder als graphische Oberfläche(in der weiteren Dokumentation wird hierauf als GUI für Graphical User Interface, referenziert) zur Verfügung stehen, oder aber als ein Command Line Interface (in der weiteren Dokumentation wird hierauf als CLI referenziert). Auch wird ein Client zur Verfügung stehen von welchen man aus von anderen Geräten auf das Programm zugreifen kann um kleine Änderungen vorzunehmen, welche unter Umständen als Antwort auf eine E-Mail welches die Software von sich aus in regelmäßigen Abständen oder Notsituationen verschicken soll, notwendig sein können. Das fertige Projekt wird für Linux sowie Windows unterstützt. Für die Realisierung wird die Sprache C# mit dem .NET Framework für Windows und mono für Linux verwendet. Es wird als Endprodukt eine Installationsfile geben welche zwei Icons für entweder CLI oder GUI erstellt sowie die Datenbank welche zur Abspeicherung von Ereignissen dient und einer XML Datei welche die vom Administrator angewendeten Konfigurationen speichert. Nun soll noch etwas näher auf die einzelnen Gebiete eingegangen werden.

1.1 Die Wahl der Sprache

C# ist eine relativ abstrakte, objektorientierte Sprache, welche heutzutage weite Verbreitung gefunden hat. Insbesondere bei Anwendungen wie Spielen aber auch besonders im Netzwerk wird das von Microsoft entwickelte .Net Framework sehr geschätzt. Sie ermöglicht schnelles Vorankommen bei der Programmierung und ist außerdem nicht so schwer zu erlernen wie systemnähere Sprachen wie C und C++. All dies, sowie die Tatsache, dass einige Mitglieder des Teams bereits viele Erfahrungen mit dieser Sprache sammeln konnten, ließ diese Sprache schlussendlich zu der Sprache der Wahl werden.

1.2 HostDetection

Diese Funktionalität wird es ermöglichen auch in angeschlossenen Netzen zu erkennen ob Hosts verfügbar sind, sofern sie ein Ping Packet nicht blockieren.

1.3 Ping

Auch wird das klassische Überprüfungspacket verwendet um, nachdem die einzelnen Hosts bekannt sind, zu überprüfen ob diese (noch) verfügbar sind. So kann man erkennen, wann ein Host das Netz auch wieder verlässt.

1.4 Dienste

Zur Überprüfung wichtiger Dienste, werden explizite Packete an diese Server geschickt und auf Antworten gewartet. Sollten diese Antworten kommen kann man stark davon ausgehen, dass diese Dienste auch verfügbar sind. Für unwichtigere Dienste wird es einen reinen Portscan geben welcher nur nachschaut ob die Ports geöffnet sind oder nicht. Das birgt zwar die Gefahr, dass sich hinter den offenen Ports ein nicht funktionierender Dienst, zB ein sogenannter Zombie, befindet, ist jedoch auch deutlich schneller als jeden einzelnen Dienst gesondert abzufragen.

1.5 Kommunikation innerhalb des Programms

Damit die vielen kleinen Einzelteile des Programmes zusammenspielen können, werden sie über eine Art Handler Klasse verbunden. Über diese Klasse sind die Programmteile dann in der Lage miteinander zu kommunizieren. Sie liefert also sozusagen einen Dolmetscher welche die verschiedenen Ausgaben der einzelnen Teile den jeweils anderen Verständlich macht.

1.6 Jobsystem

Um die Dienste welches unser Programm anbieten wird (die zuvor bereits genannten zB Ping, Hostdetection, Dienste, Notification, ...) übersichtlich und einfach zu verwalten, wird ein System einprogrammiert welches sie als 'Jobs' behandelt und so dem User einfacher zugänglich macht.

1.7 CLI

Das Command Line Interface bietet eine Möglichkeit für erfahrenere Benutzer oder User die es gewohnt sind mit der Command Line zu arbeiten, nicht immer eine Grafische Oberfläche starten zu müssen sondern direkt in der Konsole zu arbeiten. Dies ist auch ein großer Vorteil für User welche mit einem Linux Devirat arbeiten welches nicht sofort eine Oberfläche startet.

1.8 GUI

Für diejenigen welche jedoch gerne alles etwas bunter und benutzerfreundlicher haben wollen wird auch eine GUI eingebaut. Diese wird dann auch die Daten aufbereiten welche in der CLI nur in Textform verfügbar sind, und sie als Graphen ausgeben.

1.9 Speicherung und Rückblick

Die Graphen müssen logischerweise auch Werte zum Zeichnen haben, welche sie aus einer Datenbank ziehen werden. Somit ist ein gewisser Rückblick auf die Vergangenheit des Netzes gewährleistet. So kann man die Ausfallsicherheit bestimmter Komponenten gut bestimmen und ähnlich wichtige Überblicke über das Netzwerk erlangen. Die Speicherung des Rückblickes wird in einer SQLite Datenbank gespeichert da diese leicht zu bedienen ist und auch auf allen unseren unterstützten Plattformen funktioniert. Desweiteren ist SQLite Opensource und gratis.

1.10 Benachrichtigung

Um auf kritische Situationen schnellstmöglich reagieren zu können gibt es das Benachrichtigungssystem. Dieses schickt eine e-mail an den Administrator um ihm die Möglichkeit zu geben Fehler zu beheben. Auch liefert es in regelmäßigen einstellbaren Zeiten einen Bericht über die Arbeit des Netzes.

1.11 Fernzugriff

Das man das Netzwerk des öfteren auch von einem entfernten Ort verwalten will, wird es die Möglichkeit geben über eine RemoteCLI auf die Verwaltung zuzugreifen. Somit kann man auf einen Notfallbericht des Benachrichtigungssystemes auch aus der Ferne reagieren.

Kapitel 2

Pflichtenheft

Die erste Deadline stellt der 9. August dar. An diesem Tag sollen alle Einzelteile zusammenspielen und funktionieren können. Die restliche Zeit wird in Verbesserungsarbeit gesteckt.
Die Arbeitsaufteilung lautet wie folgt:

- Porcic Alin
  - Handler
  - Jobsystem
  - Ping
  - CLI
  - RemoteCLI
- Ranalter Daniel
  - HostDetection
  - Dienste
- Singh Manpreet
  - Benachrichtigung
  - GUI
- Stojanovic Marko
  - Speicherung
  - GUI