1.0 Diagnoseansatz – Systematische Fehleranalyse

Zweck dieser Seite

Diese Seite beschreibt die grundlegende Denk- und Arbeitsweise bei der professionellen IT-Fehleranalyse.

Sie ist die Einführungsseite des Kapitels. Konkrete Befehle, Befundformulare und detaillierte Diagnoseabläufe folgen auf den nächsten Seiten.

Was bedeutet systematische Fehleranalyse?

Systematische Fehleranalyse bedeutet, eine Störung nicht durch zufälliges Ausprobieren zu bearbeiten.

Stattdessen wird:

1. das sichtbare Problem beschrieben;
2. der betroffene Bereich eingegrenzt;
3. der technische Datenpfad verstanden;
4. eine mögliche Ursache als Hypothese formuliert;
5. die Hypothese mit einem gezielten Test geprüft;
6. nur die nachgewiesene Ursache behoben;
7. die vollständige Funktion anschließend verifiziert.

Das Ziel ist nicht nur, das Symptom kurzfristig zu beseitigen. Die tatsächliche technische Ursache soll gefunden, behoben und dokumentiert werden.

Ein Symptom ist keine Ursache

Eine Störung wird zunächst meistens nur als Symptom wahrgenommen.

Beispiele:

Symptom	Mögliche Ursachen
„Das Internet funktioniert nicht“	Interface, Kabel, VLAN, DHCP, Gateway, DNS, Proxy, Firewall, VPN oder Provider
„Der Server ist langsam“	CPU, RAM, Datenträger, Netzwerk, DNS, Anwendung, Datenbank, Backup oder Hypervisor
„Die Anmeldung funktioniert nicht“	Konto, Passwort, DNS, Uhrzeit, Kerberos, Domain Controller, MFA oder Berechtigungen

Symptom	Mögliche Ursachen
„Die Webseite ist nicht erreichbar“	DNS, Routing, Firewall, TCP-Port, TLS, Proxy, Webserver oder Anwendung
„Der Container startet nicht“	Image, Architektur, Konfiguration, Port, Volume, Rechte, Abhängigkeit oder Ressourcenlimit

Aus einem Symptom dürfen deshalb nicht sofort konkrete Reparaturmaßnahmen abgeleitet werden.

Typische Denkfehler

Vorschnelle Festlegung

Nach dem ersten Hinweis wird sofort eine Ursache angenommen.

Beispiel:

> „Der Hostname funktioniert nicht. Also ist der DNS-Server ausgefallen.

Der DNS-Server könnte erreichbar sein und trotzdem einen falschen Record liefern. Ebenso könnten DNS-Suffix, Clientcache, hosts-Datei, VPN-DNS oder Split-DNS die Ursache sein.

Bestätigungsfehler

Es werden nur noch Informationen gesucht, welche die eigene Vermutung bestätigen. Widersprechende Befunde werden ignoriert.

Korrelation mit Ursache verwechseln

Ein Fehler tritt nach einem Update auf. Daraus wird geschlossen, dass das Update die Ursache sein muss.

Die zeitliche Nähe ist ein wichtiger Hinweis, aber noch kein Nachweis.

Nur die eigene Komponente untersuchen

Der Clientadministrator untersucht nur den Client, der Netzwerkadministrator nur den Switch und der Serveradministrator nur den Server.

Eine Verbindung funktioniert jedoch nur, wenn der gesamte Datenpfad funktioniert.

Erfolgreichen Einzeltest überbewerten

Ein erfolgreicher Ping beweist nicht, dass DNS, TCP-Port, TLS, Authentifizierung oder Anwendung funktionieren.

Mehrere Änderungen gleichzeitig durchführen

Werden gleichzeitig DNS geändert, Firewallregeln angepasst und Dienste neu gestartet, kann anschließend nicht mehr festgestellt werden, welche Maßnahme tatsächlich wirksam war.

Neustart mit Fehlerbehebung verwechseln

Ein Neustart kann einen fehlerhaften Zustand vorübergehend beseitigen. Die Ursache ist dadurch nicht automatisch bekannt oder dauerhaft behoben.

Grundlegender Diagnosekreislauf

`Symptom → Eingrenzung → Datenpfad → Hypothese → Test → Befund → Korrektur → Verifikation → Dokumentation`

Nach jedem Test wird entschieden:

- Hypothese bestätigt;
- Hypothese widerlegt;
- Ergebnis noch nicht eindeutig;
- weiterer Test erforderlich.

Erst eine bestätigte Hypothese führt zu einer konkreten Korrektur.

Die wichtigsten Eingrenzungsmethoden

Methode	Grundidee	Typischer Einsatz
Bottom-up	Von Hardware und Verbindung zur Anwendung arbeiten	Link-, Kabel-, VLAN- und Netzwerkprobleme
Top-down	Bei Benutzer und Anwendung beginnen	Anwendungs-, Zugriffs- und Authentifizierungsfehler
Divide and Conquer	Datenpfad in Abschnitte teilen	Lange oder komplexe Verbindungswege
Follow the Path	Anfrage und Antwort vollständig verfolgen	Routing, Firewall, NAT, Proxy und VPN
Known Good Comparison	Mit funktionierendem System vergleichen	Einzelne Benutzer, Clients oder Ports
Controlled Swap	Komponente kontrolliert austauschen	Kabel, Port, SFP, Netzteil oder Client
Two-Sided Analysis	Sender und Empfänger gleichzeitig untersuchen	Paketverlust und Kommunikationsabbrüche

Methode	Grundidee	Typischer Einsatz
Historical Comparison	Aktuellen Zustand mit früherem Zustand vergleichen	Änderungen, Updates und Performanceprobleme

Keine Methode ist grundsätzlich immer die richtige. Die Auswahl hängt vom Symptom und vom bekannten Datenpfad ab.

Technischen Datenpfad betrachten

Ein Dienst besteht häufig aus mehreren voneinander abhängigen Komponenten.

Beispiel:

`Benutzer → Client → Netzwerkinterface → Switch oder Access Point → VLAN → Gateway → Firewall → Server → Anwendung → Datenbank`

Für jede Komponente muss geklärt werden:

- Erreicht die Anfrage diese Stelle?
- Wird die Anfrage angenommen?
- Wird sie korrekt weitergeleitet?
- Wird eine Antwort erzeugt?
- Kommt die Antwort beim Absender an?

Dadurch lässt sich der fehlerhafte Abschnitt schrittweise eingrenzen.

Sender, Empfänger, Hinweg und Rückweg

Eine vollständige Verbindungsanalyse betrachtet immer vier Bereiche:

Bereich	Fragestellung
Sender	Wird die Anfrage korrekt erzeugt und gesendet?
Hinweg	Erreicht die Anfrage das Ziel?
Empfänger	Nimmt das Ziel die Anfrage an und antwortet es?
Rückweg	Erreicht die Antwort den ursprünglichen Sender?

Eine Anfrage kann den Server erreichen, obwohl die Antwort wegen einer fehlenden Rückroute, einer Firewall oder asymmetrischem Routing verloren geht.

Vergleich mit einem funktionierenden System

Ein bekannt funktionierendes Vergleichssystem ist häufig aussagekräftiger als eine allgemeine Vermutung.

Verglichen werden können:

- Benutzer;
- Client;
- Betriebssystemversion;
- Netzwerkkonfiguration;
- Switchport;
- VLAN;
- DNS-Server;
- Routingtabelle;
- Gruppenmitgliedschaften;
- Richtlinien;
- Zertifikate;
- Softwareversionen;
- Konfigurationsdateien.

Entscheidend ist, möglichst nur einen Unterschied gleichzeitig zu untersuchen.

Beweise vor Veränderungen sichern

Viele technische Informationen sind flüchtig.

Dazu gehören:

- aktive Verbindungen;
- Arbeitsspeicherinhalt;
- Prozesse;
- temporäre Fehlerzustände;
- Interface-Counter;
- Routing- und Nachbartabellen;
- nicht gespeicherte Logs;
- Paketmitschnitte;
- Cluster- und Sessionzustände.

Deshalb gilt:

> “ Erst den ursprünglichen Zustand erfassen, danach verändern.

Welche Informationen konkret gesichert werden, wird auf der Seite `1.4 Flüchtige Informationen und Beweise sichern` behandelt.

Eine Änderung pro Test

Jede Änderung muss eine konkrete Hypothese prüfen.

Beispiel:

> Hypothese: Der Client befindet sich im falschen VLAN.

Geeigneter Test:

> Switchport-Zuweisung und tatsächlich gelerntes VLAN mit einem funktionierenden Port vergleichen.

Erst wenn die Hypothese bestätigt ist, wird die VLAN-Konfiguration korrigiert.

Mehrere gleichzeitige Änderungen verhindern eine eindeutige Ursachenbestimmung.

Fehlerbehebung und Verifikation trennen

Eine ausgeführte Änderung ist noch keine bestätigte Lösung.

Nach der Korrektur muss geprüft werden:

- Ist das ursprüngliche Symptom verschwunden?
- Funktioniert der tatsächliche Dienst?
- Funktioniert die Verbindung in beide Richtungen?
- Sind andere Benutzer oder Systeme weiterhin funktionsfähig?
- Enthalten die Logs neue Fehler?
- Befinden sich Monitoringwerte wieder im Normalbereich?
- Wurden temporäre Diagnoseänderungen entfernt?

Ein Fehler gilt erst dann als behoben, wenn die ursprüngliche Funktion vollständig getestet wurde.

Dokumentation als Teil der Fehlerbehebung

Eine vollständige Fehlerdokumentation enthält:

- sichtbares Symptom;
- betroffene Systeme und Benutzer;
- Beginn und Dauer;
- technische Ursache;
- durchgeführte Tests;
- eindeutige Befunde;

- ausgeführte Korrektur;
- mögliche Auswirkungen;
- Rollback;
- abschließende Verifikation;
- Präventionsmaßnahme;
- relevante Herstellerquelle.

Die Dokumentation verhindert, dass derselbe Fehler später erneut vollständig untersucht werden muss.

Grundregeln

- Erst eingrenzen, dann verändern.
- Ein Symptom ist keine Ursache.
- Vermutungen werden als Hypothesen behandelt.
- Ein einzelner erfolgreicher Test beweist nicht den gesamten Dienst.
- Sender, Empfänger, Hinweg und Rückweg müssen betrachtet werden.
- Vergleichssysteme müssen möglichst ähnlich sein.
- Nur eine kontrollierte Änderung gleichzeitig durchführen.
- Vor Veränderungen flüchtige Informationen sichern.
- Eine Änderung ist erst nach erfolgreicher Verifikation eine Lösung.
- Neustart und Reset ersetzen keine Ursachenanalyse.
- Sicherheitsfunktionen dürfen nicht unkontrolliert umgangen werden.
- Jede dauerhafte Korrektur benötigt Dokumentation und Prävention.

Seiten dieses Kapitels

Quellen

- IBM – Systematischer Troubleshooting-Ansatz
- Microsoft – Windows Networking Troubleshooting
- Microsoft – Windows Performance Troubleshooting

- Cisco – Switchport- und Interfaceprobleme
- Wireshark User’s Guide
- NIST – Incident Response Recommendations
