6.6 Fehler tritt nur manchmal auf – sporadische Störungen systematisch erfassen und korrelieren

Sporadische Fehler gehören zu den schwierigsten Störungen in der IT-Fehleranalyse. Zum Zeitpunkt der Untersuchung funktioniert das betroffene System häufig wieder normal. Einzelne manuelle Tests liefern dann keine auffälligen Ergebnisse.

Typische Aussagen sind:

- „Das passiert nur manchmal.“
- „Nach einigen Minuten geht es wieder.“
- „Heute Morgen war alles langsam.“
- „Die Verbindung bricht unregelmäßig ab.“
- „Nach einem Neustart funktioniert es wieder.“
- „Der Fehler tritt nur bei bestimmten Benutzern auf.“
- „In den Protokollen ist nichts zu sehen.“
- „Es passiert meistens nachts oder unter hoher Last.“

Eine belastbare Untersuchung benötigt deshalb:

1. einen möglichst genauen Fehlerzeitpunkt,
2. eine reproduzierbare Beschreibung des Symptoms,
3. kontinuierliche oder ereignisgesteuerte Messungen,
4. ausreichend lange aufbewahrte Protokolle,
5. synchronisierte Systemzeiten,
6. Messdaten vor, während und nach dem Fehler,
7. eine gemeinsame Zeitleiste aller beteiligten Systeme.

Ein unauffälliger Einzeltest beweist lediglich, dass der Fehler während dieses Tests nicht beobachtet wurde.

---

1. Was ist ein sporadischer Fehler?

Ein sporadischer Fehler tritt nicht dauerhaft und häufig nicht zuverlässig reproduzierbar auf.

Mögliche Erscheinungsformen:

- zufällig wirkende Verbindungsabbrüche,
- zeitweise sehr hohe Latenz,
- gelegentliche Anmeldefehler,
- nur einzelne fehlgeschlagene DNS-Anfragen,
- kurzzeitige Serverausfälle,
- unregelmäßige WLAN-Unterbrechungen,
- vereinzelte Zeitüberschreitungen,

- Dienste reagieren vorübergehend nicht,
- Dateien können manchmal nicht geöffnet werden,
- eine Anwendung verliert gelegentlich ihre Sitzung,
- ein Switchport wechselt kurzzeitig den Linkstatus,
- ein DHCP-Lease wird gelegentlich nicht erneuert,
- Zertifikats- oder Tokenfehler treten nur zu bestimmten Zeiten auf.

Sporadisch bedeutet nicht automatisch zufällig. Häufig besitzt der Fehler ein Muster, das erst durch ausreichend lange Beobachtung sichtbar wird.

---

2. Typische Ursachen

Mögliche Ursachen sind:

- kurzzeitige Link-Unterbrechung,
- beschädigtes Kabel oder loser Steckverbinder,
- instabile Stromversorgung,
- Überhitzung,
- WLAN-Interferenz,
- Roaming zwischen Access Points,
- ausgelasteter Uplink,
- Microbursts,
- kurzfristig volle Warteschlangen,
- DHCP-Lease-Erneuerung,
- ablaufender DNS-Cache oder DNS-TTL,
- Token- oder Sitzungsablauf,
- Idle-Timeout,
- VPN-Rekeying,
- Zertifikatsablauf oder fehlerhafte Zertifikatsprüfung,
- geplantes Backup,
- Virenscan,
- Snapshot,
- Replikation,
- Datenbankwartung,
- Softwareverteilung,
- Update oder Neustart,
- Ressourcenerschöpfung,
- Speicherleck,
- kurzzeitig hohe CPU- oder Datenträgerlast,
- begrenzter Connection Pool,
- Port- oder Socket-Erschöpfung,
- Paketverlust,
- Firewall- oder Proxy-Timeout,
- Load-Balancer-Health-Check,
- Cluster-Failover,

- Providerstörung,
- Fehler in einem externen Dienst.

---

3. Auswirkungen bestimmen

Vor der technischen Untersuchung muss der Umfang eingegrenzt werden.

Zu klären ist:

- ein Benutzer oder mehrere Benutzer,
- ein Client oder mehrere Clients,
- ein Standort oder mehrere Standorte,
- WLAN oder Ethernet,
- ein VLAN oder mehrere VLANs,
- ein Server oder mehrere Server,
- eine Anwendung oder mehrere Anwendungen,
- interne oder externe Ziele,
- IPv4 oder IPv6,
- mit oder ohne VPN,
- nur Upload oder nur Download,
- nur ein bestimmter Geschäftsprozess,
- nur eine bestimmte Uhrzeit,
- nur unter Last,
- nur nach längerer Inaktivität,
- nur nach Anmeldung, Aufwachen oder Netzwerkwechsel.

Eine Beobachtung wie „Das Netzwerk war kurz weg“ ist noch keine ausreichende Fehlerbeschreibung.

---

4. Fehlerzeitpunkt exakt erfassen

Der wichtigste Ausgangspunkt ist ein möglichst genauer Zeitstempel.

Ungeeignet:

```
Der Fehler war heute Morgen.
```

Besser:

```
02.08.2026 zwischen 09:14:20 und 09:15:05 Uhr
```

Noch besser:

02.08.2026, 09:14:37 Uhr:
Beim Öffnen von \\fileserver.example.test\projekte erschien nach etwa 30 Sekunden die Meldung „Der Netzwerkname wurde nicht gefunden“.

09:15:05 Uhr:
Ein erneuter Versuch war erfolgreich.

Zusätzlich zu dokumentieren:

- Zeitzone,
- Benutzer,
- Clientname,
- Client-IP,
- MAC-Adresse,
- Standort,
- Verbindungstyp,
- Access Point oder Switchport,
- Zielname,
- Ziel-IP,
- Anwendung,
- verwendeter Port,
- genaue Meldung,
- Dauer,
- Aktion unmittelbar vor dem Fehler,
- Zustand nach dem Fehler.

Screenshots sollten nach Möglichkeit die Uhrzeit enthalten. Andernfalls muss der Zeitpunkt separat dokumentiert werden.

---

5. Systemzeiten kontrollieren

Daten verschiedener Systeme können nur sinnvoll korreliert werden, wenn ihre Uhren ausreichend genau synchronisiert sind.

Zu prüfen sind:

- Uhrzeit,
- Datum,
- Zeitzone,
- konfigurierte Zeitquelle,
- Synchronisationsstatus,
- Abweichung zur Referenz,
- Zeitpunkt der letzten Synchronisation,
- Erreichbarkeit der Zeitquelle,

- Zeitstatus virtueller Maschinen,
- Zeitquelle von Netzwerkgeräten,
- Zeitquelle von Containern und Hosts.

Windows

```
Get-Date
Get-TimeZone
w32tm /query /status
w32tm /query /source
w32tm /query /configuration
```

Linux

```
date --iso-8601=seconds
timedatectl status
timedatectl timesync-status
```

macOS

```
date
systemsetup -gettimezone
systemsetup -getusingnetworktime
systemsetup -getnetworktimeserver
```

Ein Zeitunterschied von mehreren Minuten kann dazu führen, dass zusammengehörige Ereignisse in Client-, Server-, Firewall- und Netzwerkprotokollen nicht erkannt werden.

Die Uhrzeit darf nicht unkontrolliert verändert werden. In Domänen, Clustern, Datenbanken und authentifizierten Umgebungen können Zeitsprünge zusätzliche Fehler verursachen.

---

6. Fehlermeldung vollständig sichern

Zu erfassen sind:

- vollständiger Meldungstext,
- Fehlercode,
- Event-ID,
- HTTP-Statuscode,
- Anwendungscode,
- Quell- und Zielsystem,

- betroffene Datei oder Ressource,
- Zeitpunkt,
- vorherige Benutzeraktion,
- Dauer bis zur Meldung,
- Verhalten bei Wiederholung.

Beispiel:

Ungeeignet:
„VPN ging nicht.“

Geeignet:
„Am 02.08.2026 um 11:42:18 Uhr brach die bestehende VPN-Verbindung nach ungefähr 27 Minuten Laufzeit ab. Der Client meldete Fehlercode 809. Der lokale Internetzugang funktionierte währenddessen weiter. Die erneute VPN-Verbindung war um 11:43:02 Uhr erfolgreich.“

Die originale Meldung darf nicht nur sinngemäß wiedergegeben werden.

---

7. Muster suchen

Sporadische Fehler sollten nach wiederkehrenden Merkmalen untersucht werden.

Muster	möglicher Untersuchungsbereich
immer zur gleichen Uhrzeit	Backup, Scan, Update, Wartungsjob oder Provider
nach längerer Inaktivität	Idle-Timeout, Energiesparmodus oder Sitzung
nach Ablauf einer festen Dauer	Lease, Token, Session, NAT, Firewall oder VPN-Rekeying
nur unter Last	Ressourcen, Queue, Uplink, Storage oder Connection Pool
nur montags oder nach dem Wochenende	Lease, Kennwort, Zertifikat, Patch oder Neustart
nur bei einem Benutzer	Benutzerprofil, Berechtigung, Token oder Endgerät
nur bei einem Client	Hardware, Treiber, Kabel, WLAN oder lokale Software
nur an einem Standort	WAN, WLAN, Switch, Strom oder Provider
nur bei WLAN	Interferenz, Roaming, Airtime oder Access Point
nur bei VPN	Tunnel, Gateway, MTU, Rekeying oder Route
nach Aufwachen des Clients	Treiber, DHCP, DNS, VPN oder Energiesparmodus
nach Konfigurationsänderung	fehlerhafte Änderung oder Abhängigkeit
alle 30 oder 60 Minuten	geplanter Timer, Lease, Token, Job oder Polling

Muster	möglicher Untersuchungsbereich
bei hoher Raumtemperatur	Kühlung, Transceiver, Netzteil oder Hardware
bei Regen oder Wind	Außenleitung, Funkstrecke, Strom oder Providerpfad

Eine zeitliche Übereinstimmung ist zunächst nur eine Korrelation und noch kein Beweis für die Ursache.

---

8. Kontinuierliche Erreichbarkeitsmessung unter Windows

Ein einfacher protokollierter Test kann prüfen, ob ein Ziel zum Störungszeitpunkt erreichbar war.

```
$Target = "<ziel>"
$LogFile = ".\availability.csv"

"Time,Target,Status,LatencyMs" | Set-Content -Path $LogFile

while ($true) {
    $Time = Get-Date -Format "yyyy-MM-ddTHH:mm:ss.fffK"
    $Result = Test-Connection -TargetName $Target -Count 1 -ErrorAction SilentlyContinue

    if ($Result) {
        "$Time,$Target,Success,$($Result.Latency)" |
            Add-Content -Path $LogFile
    }
    else {
        "$Time,$Target,Failure," |
            Add-Content -Path $LogFile
    }

    Start-Sleep -Seconds 5
}
```

Vor dem Einsatz ist zu prüfen, welche Eigenschaften `Test-Connection` in der verwendeten PowerShell-Version zurückgibt.

Beenden:

```
[CTRL] + [C]
```

Die Messung zeigt nur die ICMP-Erreichbarkeit des gewählten Ziels. Sie beweist nicht, dass DNS, TCP, TLS oder die Anwendung funktioniert haben.

---

9. Kontinuierliche Erreichbarkeitsmessung unter Linux und macOS

```
target="<ziel>"
logfile="./availability.log"

while true; do
    printf '%s ' "$(date '+%Y-%m-%dT%H:%M:%S%z')" >> "$logfile"

    if ping -c 1 -W 2 "$target" >> "$logfile" 2>&1; then
        printf 'STATUS=SUCCESS\n' >> "$logfile"
    else
        printf 'STATUS=FAILURE\n' >> "$logfile"
    fi

    sleep 5
done
```

Je nach Betriebssystem unterscheiden sich die Optionen und die Einheit des Ping-Timeouts. Der konkrete Befehl muss auf dem Zielsystem geprüft werden.

Beenden:

```
[CTRL] + [C]
```

Sinnvolle parallele Ziele:

1. Loopback-Adresse,
2. eigene IP-Adresse,
3. Standardgateway,
4. interner Server,
5. DNS-Server,
6. VPN-Gateway,
7. externes Vergleichsziel,
8. tatsächlich betroffener Dienst.

Dadurch kann erkannt werden, ab welchem Teil des Pfades der Fehler sichtbar wird.

---

10. Dienst statt nur Ping überwachen

Wenn die Anwendung einen bestimmten TCP-Port benötigt, sollte zusätzlich dieser Port getestet werden.

Windows

```
$Target = "<ziel>"
$Port = <port>
$LogFile = ".\tcp-check.csv"

"Time,Target,Port,Success" | Set-Content -Path $LogFile

while ($true) {
    $Time = Get-Date -Format "yyyy-MM-ddTHH:mm:ss.fffK"
    $Result = Test-NetConnection -ComputerName $Target -Port $Port `
        -InformationLevel Quiet

    "$Time,$Target,$Port,$Result" |
        Add-Content -Path $LogFile

    Start-Sleep -Seconds 10
}
```

Linux und macOS

```
while true; do
    timestamp="$(date '+%Y-%m-%dT%H:%M:%S%z')"

    if nc -z -w 3 <ziel> <port>; then
        printf '%s TARGET=%s PORT=%s STATUS=SUCCESS\n' \
            "$timestamp" "<ziel>" "<port>"
    else
        printf '%s TARGET=%s PORT=%s STATUS=FAILURE\n' \
            "$timestamp" "<ziel>" "<port>"
    fi

    sleep 10
done >> tcp-check.log 2>&1
```

Ein erfolgreicher TCP-Verbindungsaufbau beweist noch nicht, dass Anmeldung, Datenbank, Backend oder vollständiger Geschäftsprozess funktionieren.

11. DNS kontinuierlich prüfen

Sporadische Namensauflösungsfehler können durch einen einzelnen Ping auf eine IP-Adresse übersehen werden.

Windows

```
$Name = "<zielname>"
$LogFile = ".\dns-check.log"

while ($true) {
    $Time = Get-Date -Format "yyyy-MM-ddTHH:mm:ss.fffK"
    "===== $Time =====" | Add-Content -Path $LogFile

    Resolve-DnsName -Name $Name -ErrorAction Continue |
        Out-String |
        Add-Content -Path $LogFile

    Start-Sleep -Seconds 30
}
```

Linux und macOS

```
while true; do
    printf '===== %s =====\n' "$(date '+%Y-%m-%dT%H:%M:%S%z')"
    dig <zielname>
    sleep 30
done >> dns-check.log 2>&1
```

Zu vergleichen sind:

- Antwortzeit,
- verwendeter DNS-Server,
- Antwortcode,
- A-Record,
- AAAA-Record,
- CNAME,
- TTL,
- wechselnde Zieladressen,
- Antwort bei Erfolg und Fehler.

Ein absichtlich wechselnder DNS-Eintrag, beispielsweise bei einem Load Balancer, darf nicht automatisch als Fehler bewertet werden.

---

12. HTTP- und Anwendungszeiten aufzeichnen

Für HTTP- oder HTTPS-Dienste kann `curl` einzelne Zeitabschnitte protokollieren.

```
while true; do
    timestamp="$(date '+%Y-%m-%dT%H:%M:%S%z')"

    result="$(curl -sS -o /dev/null \
      --connect-timeout 10 \
      --max-time 30 \
      -w 'code=%{http_code} remote=%{remote_ip} dns=%{time_namelookup} tcp=%{time_connect}
tls=%{time_appconnect} first_byte=%{time_starttransfer} total=%{time_total}' \
      https://<zielname>/ 2>&1)"

    printf '%s %s\n' "$timestamp" "$result"
    sleep 30
done >> http-check.log
```

Mögliche Einordnung:

Beobachtung	möglicher Bereich
DNS-Zeit erhöht	Resolver oder DNS-Pfad
TCP-Aufbau erhöht	Netzwerkpfad, Paketverlust oder Ziel
TLS-Zeit erhöht	Zertifikatsprüfung, Inspection oder Server
erstes Byte verspätet	Anwendung, Proxy, Backend oder Datenbank
Gesamtzeit erhöht	Übertragung, Durchsatz, Server oder Client
wechselnde Ziel-IP mit Fehler nur bei einer IP	Load Balancer oder einzelner Backend-Pfad
HTTP-Fehlercode	Anwendung, Proxy, Authentifizierung oder Backend

Die Abfrage muss auf einen ungefährlichen und autorisierten Endpunkt begrenzt werden.

---

13. Interfacezustand und Fehlerzähler regelmäßig erfassen

Sporadische Linkfehler sind häufig nur als kurzzeitiger Statuswechsel oder steigender Fehlerzähler sichtbar.

Windows

```
$LogFile = ".\adapter-statistics.csv"

while ($true) {
    $Time = Get-Date -Format "yyyy-MM-ddTHH:mm:ss.fffK"

    Get-NetAdapterStatistics |
        Select-Object @{
            Name = "Time"
            Expression = { $Time }
        }, Name, ReceivedBytes, SentBytes,
           ReceivedUnicastPackets, SentUnicastPackets,
           ReceivedDiscardedPackets, OutboundDiscardedPackets,
           ReceivedPacketErrors, OutboundPacketErrors |
        Export-Csv -Path $LogFile -Append -NoTypeInformation

    Start-Sleep -Seconds 30
}
```

Linux

```
while true; do
    printf '===== %s =====\n' "$(date '+%Y-%m-%dT%H:%M:%S%z')"
    ip -s link show dev <interface>
    ethtool <interface>
    ethtool -S <interface>
    sleep 30
done >> interface-check.log 2>&1
```

macOS

```
while true; do
    printf '===== %s =====\n' "$(date '+%Y-%m-%dT%H:%M:%S%z')"
    ifconfig <interface>
    netstat -ib
    sleep 30
done >> interface-check.log 2>&1
```

Zu beobachten sind insbesondere:

- Link up/down,
- ausgehandelte Geschwindigkeit,
- Duplex,
- CRC- oder FCS-Fehler,
- Eingabe- und Ausgabefehler,
- verworfene Pakete,
- Queue Drops,
- Carrier-Fehler,
- Link-Flaps,
- neu ausgehandelte Verbindung,
- steigende Fehlerdifferenz zwischen zwei Messzeitpunkten.

Absolute Zählerstände reichen nicht aus. Entscheidend ist, ob und wann die Zähler während des Fehlers gestiegen sind.

---

14. Switchport und Access Point überwachen

Auf der Netzwerkkomponente sollten mindestens geprüft werden:

- Portstatus,
- letzter Statuswechsel,
- Anzahl Link-Flaps,
- Speed und Duplex,
- CRC-/FCS-Fehler,
- Input Errors,
- Output Errors,
- Discards,
- Queue Drops,
- Broadcast- und Multicast-Anteil,
- PoE-Status,
- Temperatur,
- Transceiverwerte,
- VLAN-Zugehörigkeit,
- Spanning-Tree-Ereignisse,
- Port-Channel-Status,
- MAC-Adresswechsel,
- WLAN-An- und Abmeldungen,
- Roaming-Ereignisse,
- Deauthentication,
- Signalstärke,
- Wiederholungsrate,
- Kanal- und Airtime-Auslastung.

Die Befehle und Zählerbezeichnungen unterscheiden sich je nach Hersteller und Modell. Ihre genaue Bedeutung muss anhand der jeweiligen Herstellerdokumentation geprüft werden.

---

15. Ereignisprotokolle unter Windows zeitlich eingrenzen

Beispiel für Systemereignisse eines bestimmten Zeitraums:

```
$Start = Get-Date "2026-08-02 09:10:00"
$End = Get-Date "2026-08-02 09:20:00"

Get-WinEvent -FilterHashtable @{
    LogName   = "System"
    StartTime = $Start
    EndTime   = $End
} |
    Select-Object TimeCreated, Id, LevelDisplayName,
                  ProviderName, Message |
    Sort-Object TimeCreated
```

Mehrere Protokolle können getrennt untersucht werden:

```
$Logs = @(
    "System",
    "Application",
    "Microsoft-Windows-DNS-Client/Operational",
    "Microsoft-Windows-WLAN-AutoConfig/Operational"
)

foreach ($Log in $Logs) {
    Get-WinEvent -FilterHashtable @{
        LogName   = $Log
        StartTime = $Start
        EndTime   = $End
    } -ErrorAction SilentlyContinue |
        Select-Object @{
            Name = "Log"
            Expression = { $Log }
        }, TimeCreated, Id, ProviderName, LevelDisplayName, Message
}
```

Nicht jedes Protokoll ist auf jedem System vorhanden oder aktiviert. Zusätzliche Protokollierung sollte kontrolliert und mit Blick auf Speicherbedarf, Datenschutz und Systemlast aktiviert werden.

---

16. Protokolle unter Linux zeitlich eingrenzen

```
journalctl \
  --since "2026-08-02 09:10:00" \
  --until "2026-08-02 09:20:00"
```

Nur Meldungen eines Dienstes:

```
journalctl \
  -u <dienst>.service \
  --since "2026-08-02 09:10:00" \
  --until "2026-08-02 09:20:00"
```

Kernelmeldungen:

```
journalctl \
  -k \
  --since "2026-08-02 09:10:00" \
  --until "2026-08-02 09:20:00"
```

Neue Meldungen fortlaufend beobachten:

```
journalctl -f
```

Zu berücksichtigen sind:

- Zeitzone,
- persistente oder flüchtige Speicherung,
- Aufbewahrungsdauer,
- Rotation,
- Rate Limiting,
- ausreichender Datenträgerspeicher,
- Berechtigungen,
- zentrale Weiterleitung.

Ein fehlender Eintrag beweist nicht automatisch, dass kein Fehler aufgetreten ist. Das Ereignis kann außerhalb der Aufbewahrungszeit liegen, in einem anderen Protokoll stehen oder nicht

protokolliert worden sein.

---

17. Protokolle unter macOS zeitlich eingrenzen

```
log show \
  --start "2026-08-02 09:10:00" \
  --end "2026-08-02 09:20:00" \
  --style syslog
```

Nach Prozess filtern:

```
log show \
  --start "2026-08-02 09:10:00" \
  --end "2026-08-02 09:20:00" \
  --predicate 'process == "<prozess>"' \
  --style syslog
```

Fortlaufende Beobachtung:

```
log stream --style syslog
```

Die genaue Syntax und die verfügbaren Protokolldaten können sich zwischen macOS-Versionen unterscheiden.

---

18. Ringpuffer-Paketmitschnitt mit Dumpcap

Ein Ringpuffer bewahrt nur die letzten Aufzeichnungsdateien auf. Wenn der Fehler auftritt, muss die Aufzeichnung zeitnah gestoppt werden, bevor die relevanten Daten überschrieben werden.

Interfaces anzeigen:

```
dumpcap -D
```

Beispiel:

```
dumpcap \
  -i <interface> \
  -w intermittent.pcapng \
```

```
  -b duration:300 \
  -b files:12
```

Dieses Beispiel:

- wechselt alle 300 Sekunden die Datei,
- hält maximal 12 Dateien,
- überschreibt anschließend die ältesten Dateien,
- bewahrt damit ungefähr die letzten 60 Minuten auf.

Alternative Begrenzung nach Dateigröße:

```
dumpcap \
  -i <interface> \
  -w intermittent.pcapng \
  -b filesize:100000 \
  -b files:10
```

Hierbei ist `filesize` in Kilobyte angegeben.

Optionaler Capture-Filter:

```
dumpcap \
  -i <interface> \
  -f "host <ziel-ip> and port <port>" \
  -w intermittent.pcapng \
  -b duration:300 \
  -b files:12
```

Capture-Filter und Wireshark-Anzeigefilter besitzen unterschiedliche Syntax. Ein zu enger Filter kann die für die Ursachenanalyse benötigten Pakete ausschließen.

---

19. Ringpuffer-Paketmitschnitt mit tcpdump

```
tcpdump \
  -i <interface> \
  -s 0 \
  -G 300 \
  -W 12 \
  -w 'intermittent-%Y%m%d-%H%M%S.pcap'
```

Möglicher Filter:

```
tcpdump \
  -i <interface> \
  -s 0 \
  -G 300 \
  -W 12 \
  -w 'intermittent-%Y%m%d-%H%M%S.pcap' \
  'host <ziel-ip> and port <port>'
```

Die Kombination und genaue Wirkung der Rotationsoptionen kann von der installierten `tcpdump`-Version abhängen und muss vor dem produktiven Einsatz mit der lokalen Handbuchseite geprüft werden:

```
man tcpdump
```

Paketaufzeichnungen können personenbezogene Daten, Anmeldedaten, Sitzungsinformationen und vertrauliche Inhalte enthalten. Sie dürfen nur mit entsprechender Berechtigung erstellt, geschützt gespeichert und nach Abschluss kontrolliert gelöscht werden.

---

20. Windows-Paketmitschnitt mit Pktmon

Filter anzeigen:

```
pktmon filter list
```

Vorhandene Filter entfernen:

```
pktmon filter remove
```

Beispiel für eine begrenzte zirkuläre Aufzeichnung:

```
pktmon start `
  --capture `
  --comp nics `
  --pkt-size 0 `
  --file-name C:\Temp\intermittent.etl `
  --file-size 512 `
```

```
  --log-mode circular
```

Aufzeichnung stoppen:

```
pktmon stop
```

In PCAPNG umwandeln:

```
pktmon etl2pcap `
  C:\Temp\intermittent.etl `
  --out C:\Temp\intermittent.pcapng
```

Im zirkulären Modus werden bei Erreichen der festgelegten Größe ältere Ereignisse überschrieben. Die Aufzeichnung muss daher nach Auftreten des Fehlers zeitnah gestoppt werden.

Die verfügbaren Optionen unterscheiden sich zwischen Windows-Versionen. Vor dem Einsatz ist die lokale Hilfe zu prüfen:

```
pktmon start help
pktmon etl2pcap help
```

---

21. Monitoringauflösung passend wählen

Ein Monitoringintervall von fünf Minuten kann einen Fehler übersehen, der nur 20 Sekunden dauert.

Beispiel:

```
09:10:00 Messung erfolgreich
09:12:15 Fehler beginnt
09:12:40 Fehler endet
09:15:00 Messung erfolgreich
```

Das Monitoring zeigt keinen Ausfall, obwohl der Dienst 25 Sekunden nicht verfügbar war.

Mögliche Maßnahmen:

- kürzeres Prüfintervall,
- aktiver Diensttest statt nur Ping,
- mehrere Messpunkte,
- zeitlich begrenzte höhere Auflösung,

- Event- oder Trap-Auswertung,
- Aufzeichnung von Minimal-, Maximal- und Durchschnittswerten,
- Perzentile statt ausschließlich Mittelwerte,
- Statuswechsel sofort protokollieren,
- Fehlerzähler als Differenz erfassen,
- Rohdaten ausreichend lange aufbewahren.

Eine höhere Auflösung erzeugt mehr Last, Daten und Warnmeldungen. Sie sollte gezielt für die betroffene Komponente und einen begrenzten Zeitraum verwendet werden.

---

22. Welche Messwerte langfristig hilfreich sind

Client

- CPU,
- RAM,
- Datenträgerlatenz,
- Netzwerkstatus,
- Linkgeschwindigkeit,
- WLAN-Signal,
- verwendeter Access Point,
- DHCP-Lease,
- DNS-Server,
- VPN-Status,
- Ereignisprotokolle.

Switch und Router

- Interface up/down,
- Fehler und Discards,
- Auslastung,
- Queue Drops,
- Port-Channel-Status,
- Routingnachbarn,
- CPU,
- RAM,
- Temperatur,
- Stromversorgung.

Firewall und VPN

- Sitzungsanzahl,
- NAT-Auslastung,
- CPU,
- RAM,
- Paketverlust,

- Drops,
- Tunnelstatus,
- Rekeying,
- Failover,
- Sicherheitsereignisse.

Server

- CPU pro Prozess,
- RAM,
- Paging oder Swapping,
- Datenträgerlatenz,
- I/O-Wait,
- Netzwerkfehler,
- offene Verbindungen,
- Queue,
- Anwendungsthreads,
- Connection Pools,
- Datenbanklocks,
- Dienstneustarts.

WLAN

- Signalstärke,
- Signal-Rausch-Verhältnis,
- Kanal,
- Airtime,
- Wiederholungsrate,
- Roaming,
- Deauthentication,
- Access-Point-Auslastung,
- Uplinkstatus.

---

23. Ablaufzeiten und regelmäßige Erneuerungen prüfen

Sporadische Fehler können entstehen, wenn ein regelmäßig erneuerter Zustand abläuft.

Zu prüfen sind:

- DHCP-Lease,
- DNS-TTL,
- ARP- und Neighbor-Cache,
- Kerberos-Ticket,
- OAuth- oder Anwendungstoken,
- Websitzung,
- Cookie,

- Firewall-Session,
- NAT-Zuordnung,
- Load-Balancer-Persistenz,
- VPN-SA und Rekeying,
- Zertifikatsgültigkeit,
- CRL- oder OCSP-Erreichbarkeit,
- Kennwortablauf,
- Maschinenkennwort,
- Connection Pool,
- Idle-Timeout,
- Keepalive,
- Lease eines verteilten Locks.

Beispiel:

```
Fehlerabstand: ungefähr 60 Minuten
Sitzungs-Timeout: 60 Minuten
```

Diese Übereinstimmung begründet eine Hypothese, beweist sie aber noch nicht. Der tatsächliche Ablauf und die anschließende Erneuerung müssen in Protokollen oder Paketdaten bestätigt werden.

---

24. Geplante Jobs und Wartungsarbeiten korrelieren

Zu untersuchen sind:

- Backuppläne,
- Snapshots,
- Virenscans,
- Patchmanagement,
- Softwareverteilung,
- Datenbankwartung,
- Logrotation,
- Replikation,
- Synchronisation,
- Indexierung,
- Cronjobs,
- systemd-Timer,
- Windows-Aufgabenplanung,
- Cloud-Automationen,
- Container-Restarts,
- Zertifikatserneuerung,
- DHCP- oder DNS-Bereinigung,
- Reportingjobs,
- geplante Providerarbeiten.

Windows

```
Get-ScheduledTask |
    Select-Object TaskPath, TaskName, State
```

Letzte und nächste Ausführung:

```
Get-ScheduledTask |
    Get-ScheduledTaskInfo |
    Select-Object TaskName, LastRunTime, LastTaskResult, NextRunTime
```

Linux

```
systemctl list-timers --all
```

Zusätzlich können je nach System Cron-Konfigurationen und deren Protokolle relevant sein.

Nur autorisierte Konfigurationen dürfen eingesehen werden. Aufgaben sollten nicht allein wegen einer zeitlichen Überschneidung deaktiviert werden.

---

25. Temperatur, Strom und physische Umgebung berücksichtigen

Kurzzeitige Ausfälle können durch physische Bedingungen verursacht werden.

Zu prüfen sind:

- Gerätetemperatur,
- Lüfterstatus,
- Netzteilstatus,
- PoE-Leistung,
- USV-Ereignisse,
- Spannungsschwankungen,
- transceiverbezogene Warnungen,
- Tür- oder Racktemperatur,
- Kabelbewegung,
- Feuchtigkeit,
- Außenleitung,
- Bauarbeiten,
- Funkstörungen,
- Neustarts nach Stromverlust.

Typisches Muster:

Hohe Last

→ Temperatur steigt

→ Schnittstelle oder Gerät fällt kurzzeitig aus

→ automatische Wiederherstellung

→ Untersuchung erfolgt später bei normaler Temperatur

Ein Neustart kann Temperatur-, Speicher- und Fehlerzustände zurücksetzen und dadurch wichtige Beweise beseitigen.

---

26. Änderungen und Deployments berücksichtigen

Zu dokumentieren sind:

- Zeitpunkt der Änderung,
- betroffene Systeme,
- vorherige Version,
- neue Version,
- geänderte Konfiguration,
- verantwortlicher Change,
- Rollbackmöglichkeit,
- erste beobachtete Störung,
- weitere abhängige Systeme.

Mögliche Änderungen:

- Treiberupdate,
- Firmwareupdate,
- Betriebssystemupdate,
- Firewallregel,
- Routingänderung,
- DNS-Änderung,
- Zertifikatswechsel,
- Anwendungsversion,
- Datenbankschema,
- Proxykonfiguration,
- Load-Balancer-Pool,
- VLAN-Änderung,
- WLAN-Kanalplanung,
- Container-Image,
- Ressourcenlimit.

„Seit dem Update“ ist eine wichtige Beobachtung, aber noch kein technischer Nachweis. Die fehlerhafte Version oder Einstellung muss durch Vergleich, Protokoll oder kontrollierten Rollback bestätigt werden.

27. Daten auf einer gemeinsamen Zeitleiste korrelieren

Beispiel:

Uhrzeit	Client	Switch	Firewall	Server
09:14:31	Anwendung startet Anfrage	Port aktiv	Sitzung angelegt	Anfrage empfangen
09:14:34	wartet auf Antwort	Output Drops steigen	Weiterleitung erfolgt	Datenträgerlatenz steigt
09:14:37	Timeout	Port weiterhin aktiv	Sitzung beendet	Anwendung antwortet verspätet
09:15:02	erneuter Versuch	Zähler stabil	neue Sitzung	Antwort normal

Erst durch die gemeinsame Zeitleiste wird sichtbar, dass der Clienttimeout mit der erhöhten Datenträgerlatenz zusammenfällt.

Zu korrelieren sind:

- Benutzeraktion,
- Clientprotokoll,
- DNS-Antwort,
- Paketmitschnitt,
- Switchport,
- Access Point,
- Firewall,
- VPN-Gateway,
- Load Balancer,
- Reverse Proxy,
- Serverbetriebssystem,
- Anwendung,
- Datenbank,
- Storage,
- Hypervisor,
- Providerereignis.

Zeitangaben müssen auf dieselbe Zeitzone oder eine eindeutig dokumentierte Referenz umgerechnet werden.

---

28. Ereignisgesteuerte Datensicherung

Wenn der Benutzer den Fehler bemerkt, sollte ein festgelegtes Verfahren ausgelöst werden.

Beispiel:

```
1. genaue Uhrzeit notieren;
2. Screenshot erstellen;
3. betroffene Aktion nicht mehrfach unkontrolliert wiederholen;
4. Diagnosemarkierung oder Ticket auslösen;
5. laufenden Ringpuffer zeitnah stoppen;
6. Clientzustand sichern;
7. relevante Server- und Netzwerkprotokolle exportieren;
8. Monitoringdaten für den Zeitraum schützen;
9. Vergleichssystem prüfen;
10. Dateien eindeutig beschriften.
```

Beispiel für eine Markierungsdatei unter PowerShell:

```
$Time = Get-Date -Format "yyyy-MM-ddTHH-mm-ss.fffK"
"Fehler durch Benutzer beobachtet: $Time" |
   Set-Content ".\incident-marker-$Time.txt"
```

Beispiel unter Bash:

```
timestamp="$(date '+%Y-%m-%dT%H-%M-%S%z')"
printf 'Fehler durch Benutzer beobachtet: %s\n' "$timestamp" \
  > "incident-marker-$timestamp.txt"
```

Der Marker liefert keine Ursache, erleichtert aber das Auffinden des relevanten Zeitraums.

---

29. Praxisfall A: Verbindung bricht ungefähr jede Stunde ab

Symptom

- eine Webanwendung trennt den Benutzer unregelmäßig,
- der Fehler tritt meistens nach längerer Nutzung auf,
- Netzwerk und Server bleiben erreichbar,
- eine erneute Anmeldung funktioniert sofort.

Prüfung

1. genaue Sitzungsdauer erfassen.
2. Browser-, Proxy- und Anwendungsprotokolle vergleichen.
3. Token- und Sitzungsablauf kontrollieren.
4. Firewall- und Load-Balancer-Timeout berücksichtigen.

5. Zeitpunkt der letzten Benutzeraktivität dokumentieren.
6. Ablauf und Erneuerung im Netzwerk- oder Anwendungsprotokoll prüfen.
7. funktionierenden Benutzer oder Client vergleichen.

Mögliche Ursache

Die Anwendung erneuert ein ablaufendes Token nicht korrekt. Nach Ablauf wird die nächste Anfrage abgelehnt.

Nachprüfung

- Sitzung bleibt über den bisherigen Fehlerzeitpunkt hinaus aktiv,
- Token wird ordnungsgemäß erneuert,
- keine zusätzlichen Anmeldefehler entstehen,
- Firewall und Netzwerk bleiben unverändert unauffällig,
- Ursache und Korrektur sind dokumentiert.

---

30. Praxisfall B: Switchport fällt kurzzeitig aus

Symptom

- ein Client verliert gelegentlich für einige Sekunden die Verbindung,
- anschließend funktioniert die Verbindung automatisch wieder,
- spätere Pingtests sind unauffällig,
- andere Clients sind nicht betroffen.

Prüfung

1. Client- und Switchzeit synchronisieren.
2. Linkstatus kontinuierlich protokollieren.
3. Switchportereignisse auswerten.
4. Fehlerzähler vor und nach dem Ereignis vergleichen.
5. Kabel und Steckverbindungen kontrollieren.
6. Energiespar- und Treibereinstellungen berücksichtigen.
7. kontrollierten Kabel- oder Portvergleich durchführen.

Mögliche Ursache

Eine beschädigte Steckverbindung verursacht kurzzeitige Link-Flaps.

Nachprüfung

- keine weiteren Link-Flaps,
- Fehlerzähler steigen nicht,
- Verbindung bleibt unter kontrollierter Last stabil,
- Vergleich über einen ausreichend langen Zeitraum,

- getauschte Komponente und Messergebnisse dokumentiert.

---

31. Praxisfall C: Dienst jeden Morgen kurz langsam

Symptom

- die Anwendung ist täglich zwischen 08:00 und 08:10 Uhr langsam,
- Netzwerkdurchsatz und Paketverlust sind unauffällig,
- Zeit bis zum ersten Byte steigt,
- danach normalisiert sich der Dienst.

Prüfung

1. Anwendungszeiten kontinuierlich erfassen.
2. Serverressourcen im betroffenen Zeitraum prüfen.
3. geplante Aufgaben und Backuppläne vergleichen.
4. Datenträgerlatenz und Datenbanklocks untersuchen.
5. Proxy- und Anwendungsprotokolle korrelieren.
6. Zustand außerhalb des Zeitfensters vergleichen.
7. Job nicht ungeprüft deaktivieren.

Mögliche Ursache

Ein täglicher Datenbankjob erzeugt hohe Datenträgerlast und konkurriert mit den Anwendungsabfragen.

Nachprüfung

- Job wird in einem vorgesehenen Wartungsfenster ausgeführt oder kontrolliert angepasst,
- Datenträgerlatenz bleibt während der Nutzungszeit normal,
- Zeit bis zum ersten Byte ist stabil,
- ursprünglicher Geschäftsprozess funktioniert,
- Backup- oder Wartungsziel bleibt erfüllt.

---

32. Praxisfall D: WLAN-Unterbrechung nur beim Standortwechsel

Symptom

- Verbindung funktioniert an festen Standorten,
- beim Wechsel zwischen zwei Gebäudebereichen entstehen kurze Unterbrechungen,
- der Client verbindet sich anschließend automatisch erneut,
- Ethernet ist nicht betroffen.

Prüfung

1. genauen Weg und Zeitpunkt dokumentieren.

2. verwendeten Access Point vor und nach dem Fehler erfassen.
3. Roaming- und Deauthentication-Ereignisse prüfen.
4. Signalstärke und Überlappung untersuchen.
5. Authentifizierungsdauer berücksichtigen.
6. Kanal- und Airtime-Auslastung prüfen.
7. Vergleich mit anderem Client durchführen.

Mögliche Ursache

Der Client hält zu lange an einem schwächer werdenden Access Point fest und wechselt verspätet zur benachbarten Funkzelle.

Nachprüfung

- Roaming erfolgt innerhalb des vorgesehenen Bereichs,
- Unterbrechungszeit ist reduziert,
- andere Clients und Funkzellen funktionieren weiterhin,
- Änderungen entsprechen der WLAN-Planung,
- Messwerte vor und nach der Änderung sind dokumentiert.

---

33. Ungeeignete Sofortmaßnahmen

Problematisch sind insbesondere:

- Systeme sofort neu starten,
- mehrere Komponenten gleichzeitig verändern,
- Protokolle erst lange nach dem Fehler prüfen,
- Benutzerzeitangaben ungeprüft übernehmen,
- einen erfolgreichen Ping als Gegenbeweis verwenden,
- Monitoringmittelwerte ohne Rohdaten bewerten,
- Protokollierung unbegrenzt aktivieren,
- Mitschnitte ohne Größenbegrenzung erstellen,
- Capture-Filter zu eng setzen,
- alte Ringpufferdateien versehentlich überschreiben,
- Firewalls oder Sicherheitsfunktionen pauschal deaktivieren,
- geplante Jobs allein wegen zeitlicher Überschneidung abschalten,
- Treiber oder Firmware ohne Vergleich austauschen,
- Uhren während der Untersuchung manuell verändern,
- vertrauliche Protokolle ungeschützt speichern,
- einen zeitlichen Zusammenhang sofort als Ursache darstellen,
- nur den betroffenen Client untersuchen,
- nur Netzwerkdaten betrachten und Server oder Anwendung ignorieren.

---

34. Vollständige Prüfreihenfolge

1. Symptom eindeutig beschreiben.
2. genaue Fehlermeldung sichern.
3. Beginn und Ende dokumentieren.
4. Zeitzone erfassen.
5. Benutzer, Client und Standort bestimmen.
6. Quelle, Ziel, Port und Anwendung dokumentieren.
7. betroffene und nicht betroffene Systeme bestimmen.
8. Häufigkeit und bisherige Zeitpunkte sammeln.
9. Muster nach Uhrzeit, Dauer und Last suchen.
10. Systemzeiten aller Beteiligten kontrollieren.
11. Vergleichssystem festlegen.
12. geeignete Messintervalle bestimmen.
13. Gateway und Ziel kontinuierlich prüfen.
14. betroffenen TCP-Dienst überwachen.
15. DNS-Antworten protokollieren.
16. Anwendungszeiten erfassen.
17. Interfacezustand und Fehlerzähler aufzeichnen.
18. Switchport oder Access Point überwachen.
19. Serverressourcen protokollieren.
20. relevante Ereignisprotokolle sichern.
21. zentrale Protokollierung und Aufbewahrung prüfen.
22. bei Bedarf autorisierten Ringpuffer-Mitschnitt starten.
23. Speicherbedarf und Datenschutz berücksichtigen.
24. Verfahren zum Markieren des Fehlerzeitpunkts festlegen.
25. bei Auftreten des Fehlers Ringpuffer sichern.
26. Daten vor, während und nach dem Ereignis vergleichen.
27. DHCP-, DNS-, Token-, Sitzungs- und Idle-Zeiten prüfen.
28. geplante Jobs, Backups, Scans und Updates korrelieren.
29. Temperatur, Stromversorgung und physische Umgebung berücksichtigen.
30. Provider- und Wartungsereignisse prüfen.
31. Änderungen und Deployments berücksichtigen.
32. gemeinsame Zeitleiste erstellen.
33. konkrete Hypothese formulieren.
34. genau eine kontrollierbare Änderung durchführen.
35. dieselben Messungen weiterlaufen lassen.
36. ausreichend lange Nachbeobachtung durchführen.
37. ursprüngliche Funktion erneut prüfen.
38. Vergleichssysteme kontrollieren.
39. temporäre Diagnosekonfiguration entfernen.
40. Ursache, Maßnahme und Nachweis dokumentieren.
41. dauerhaftes Monitoring oder Prävention festlegen.

---

35. Dokumentationsbeispiel

Ticket: INC-20804

Zeitraum der Untersuchung: 02.08.2026 bis 05.08.2026

Betroffen: Client-WS-17, Ethernet, Switch SW-F2-03, Port 18

Nicht betroffen: weitere Clients desselben VLANs

Ziel: fileserver.example.test, TCP 445

Symptom:

Der Client verliert ein- bis dreimal täglich für ungefähr 5 bis 15 Sekunden den Zugriff auf den Dateiserver. Anschließend funktioniert der Zugriff ohne Benutzeraktion wieder.

Erfasste Fehlerzeitpunkte:

- 02.08.2026, 10:17:42 bis 10:17:51 Uhr
- 03.08.2026, 14:32:08 bis 14:32:19 Uhr
- 05.08.2026, 09:06:14 bis 09:06:23 Uhr

Messungen:

- kontinuierlicher Ping zu Gateway und Dateiserver
- TCP-Prüfung auf Port 445
- Interfacezähler des Clients
- Switchportstatus und Fehlerzähler
- Windows-Systemereignisse
- begrenzter Ringpuffer-Paketmitschnitt
- Serverressourcen und SMB-Protokolle

Beobachtung:

Zum jeweiligen Fehlerzeitpunkt meldete der Switch einen kurzen Linkverlust an Port 18. Gleichzeitig wechselte das Clientinterface in den Zustand „Disconnected“. Gateway und Server waren für andere Clients erreichbar. Am Switchport stiegen zusätzlich die physischen Fehlerzähler.

Hypothese:

Die physische Verbindung zwischen Client und Switch ist instabil.

Kontrollierte Änderung:

Das Patchkabel wurde durch ein geprüftes Kabel ersetzt. Client, Switchport, Treiber und Serverkonfiguration blieben unverändert.

Nachprüfung:

- sieben Tage kontinuierliche Beobachtung
- keine weiteren Link-Flaps
- keine steigenden physischen Fehlerzähler
- Gateway und Dateiserver durchgehend erreichbar
- ursprünglicher Dateizugriff funktioniert
- andere Systeme unverändert fehlerfrei

Ursache:

Beschädigtes Patchkabel mit sporadischer Unterbrechung.

Prävention:
- Switchport-Flaps zentral überwachen
- steigende physische Fehlerzähler alarmieren
- Fehlerzeitpunkte im Ticket immer sekundengenau dokumentieren

---

36. Checkliste „Fehler tritt nur manchmal auf“

- [ ] das genaue Symptom wurde beschrieben.
- [ ] der originale Meldungstext wurde gesichert.
- [ ] Fehlercode oder Event-ID wurde dokumentiert.
- [ ] Beginn und Ende des Fehlers wurden erfasst.
- [ ] die Zeitzone wurde dokumentiert.
- [ ] Benutzer, Client und Standort sind bekannt.
- [ ] Quelle, Ziel, Port und Anwendung sind bekannt.
- [ ] betroffene und nicht betroffene Systeme wurden bestimmt.
- [ ] Häufigkeit und bisherige Zeitpunkte wurden gesammelt.
- [ ] zeitliche oder belastungsabhängige Muster wurden gesucht.
- [ ] Systemzeiten der beteiligten Systeme wurden geprüft.
- [ ] ein funktionierendes Vergleichssystem wurde festgelegt.
- [ ] Messintervalle sind kürzer als die vermutete Fehlerdauer.
- [ ] Gateway und Ziel werden kontinuierlich geprüft.
- [ ] der tatsächliche Dienst wird zusätzlich zu Ping geprüft.
- [ ] DNS wird bei passenden Symptomen protokolliert.
- [ ] Anwendungszeiten werden bei passenden Symptomen erfasst.
- [ ] Interfacezustand und Fehlerzähler werden aufgezeichnet.
- [ ] Switchport oder Access Point werden berücksichtigt.

- [ ] Serverressourcen werden erfasst.
- [ ] relevante Ereignisprotokolle sind ausreichend lange verfügbar.
- [ ] Monitoringauflösung wurde kontrolliert erhöht.
- [ ] Ringpuffer besitzt eine Größen- oder Zeitbegrenzung.
- [ ] Paketaufzeichnung erfolgt nur mit Berechtigung.
- [ ] Verfahren zum Markieren des Fehlerzeitpunkts ist festgelegt.
- [ ] Daten vor, während und nach dem Fehler wurden gesichert.
- [ ] DHCP-Lease und DNS-TTL wurden berücksichtigt.
- [ ] Token-, Session- und Idle-Timeouts wurden berücksichtigt.
- [ ] VPN-Rekeying und Firewall-Sitzungen wurden berücksichtigt.
- [ ] geplante Jobs, Backups, Scans und Updates wurden geprüft.
- [ ] Temperatur und Stromversorgung wurden berücksichtigt.
- [ ] Änderungen und Deployments wurden geprüft.
- [ ] Daten mehrerer Systeme wurden auf einer Zeitleiste korreliert.
- [ ] Korrelation und nachgewiesene Ursache wurden getrennt bewertet.
- [ ] eine konkrete Hypothese wurde formuliert.
- [ ] nur eine kontrollierbare Änderung wurde vorgenommen.
- [ ] die Nachbeobachtung war ausreichend lang.
- [ ] ursprüngliche Funktion und Vergleichssysteme wurden geprüft.
- [ ] temporäre Diagnosekonfigurationen wurden entfernt.
- [ ] Ursache, Maßnahme und Nachweis wurden dokumentiert.
- [ ] Monitoring oder Prävention wurde festgelegt.

---

37. Schnellreferenz

Beobachtung	wahrscheinlicher Untersuchungsbereich
Fehler immer zur gleichen Uhrzeit	Job, Backup, Scan, Update oder Provider
Fehler nach fester Laufzeit	Lease, Token, Session, NAT, Firewall oder VPN
Fehler nach Inaktivität	Idle-Timeout, Energiesparen oder Sitzung
nur ein Client betroffen	Client, Kabel, Treiber, Port oder Benutzerprofil
alle Clients eines Standorts betroffen	Uplink, WAN, Firewall, Strom oder Provider
nur WLAN betroffen	Funkumgebung, Roaming, Airtime oder Access Point
nur beim Standortwechsel	Roaming, Authentifizierung oder Funkabdeckung
nur unter Last	CPU, RAM, Storage, Queue, Uplink oder Pool

Beobachtung	wahrscheinlicher Untersuchungsbereich
nur nachts	Backup, Snapshot, Wartung, Scan oder Replikation
Linkstatus wechselt kurz	Kabel, Port, Netzwerkkarte, Strom oder Treiber
Fehlerzähler steigen	physischer Link, Queue, Überlastung oder Hardware
Netzwerk normal, Anwendung hängt	Anwendung, Backend, Datenbank oder Storage
nur einzelne Ziel-IP fehlerhaft	Load Balancer, Backend oder Zielpfad
nur nach Aufwachen	Treiber, DHCP, DNS, VPN oder Energiesparmodus
nur über VPN	Tunnel, Rekeying, MTU, Gateway oder Route
Fehler verschwindet nach Neustart	Ressourcen, Speicherleck, Zustand oder Dienst
Monitoring zeigt nichts	Intervall zu groß, falscher Test oder fehlende Rohdaten
Logs zeigen nichts	falsches Protokoll, Rotation, Zeitabweichung oder fehlende Erfassung
Paketmitschnitt enthält Ereignis nicht	Filter, falsches Interface oder Überschreibung
zeitlicher Zusammenhang mit Job	mögliche Korrelation, Ursache noch zu beweisen
Problem tritt bei Temperaturanstieg auf	Kühlung, Netzteil, Transceiver oder Hardware
regelmäßige DNS-Fehler	Resolver, TTL, Cache, Netzwerkpfad oder DNS-Server
regelmäßige Abmeldung	Token, Sitzung, Cookie, Proxy oder Load Balancer
kurze Ausfälle ohne Linkverlust	Dienst, Routing, Firewall, Queue oder Server
andere Clients funktionieren	gemeinsamer Server nicht automatisch ausgeschlossen, aber Clientpfad priorisieren

---

Merksatz

> Sporadische Fehler werden selten durch einen einzelnen nachträglichen Test gefunden. Entscheidend sind sekundengenaue Zeitstempel, kontinuierliche Messungen, begrenzte Ringpuffer, ausreichend lange Protokollaufbewahrung und eine gemeinsame Zeitleiste von Client, Netzwerk, Server und Anwendung. Erst wenn Messdaten den Fehler vor, während und nach einer kontrollierten Änderung zeigen, ist die Ursache belastbar nachgewiesen.

Quellen und weiterführende Dokumentation

- Microsoft Learn – Get-WinEvent
- Microsoft Learn – Test-Connection
- Microsoft Learn – Test-NetConnection

- Microsoft Learn – Get-NetAdapterStatistics
- Microsoft Learn – Resolve-DnsName
- Microsoft Learn – Windows Time Service Tools and Settings
- Microsoft Learn – Packet Monitor
- Microsoft Learn – pktmon start
- Microsoft Learn – pktmon etl2pcap
- Microsoft Learn – Windows Event Forwarding
- Wireshark – Dumpcap Manual
- Wireshark User’s Guide – Capturing with Dumpcap
- Wireshark User’s Guide – Capturing with tcpdump
- tcpdump Manual Page
- systemd – journalctl
- systemd – timedatectl
- curl – Write-out Variables
- RFC 5905 – Network Time Protocol Version 4
- RFC 2131 – Dynamic Host Configuration Protocol
- RFC 9293 – Transmission Control Protocol