LOGZILLA DOCUMENTATION

BGP Network Monitoring Correlation

Track BGP adjacency changes, calculate outage duration, and detect chronic instability in Cisco routers using LogZilla Event Correlation rules

Event Correlation · Generated April 27, 2026 · logzilla.ai/docs/event-correlation/bgp-network-monitoring-correlation

BGP Network Monitoring and Correlation

Purpose

BGP (Border Gateway Protocol) monitoring requires correlating adjacency changes, outage duration tracking, and network stability analysis. Pre- processing extracts BGP-specific fields, and SEC handles time-based correlation patterns that triggers do not cover.

Prerequisites

- Event Correlation must be enabled.

BGP Adjacency Outage Tracking

Required App: *Cisco IOS*

Forwarder Configuration for Cisco Network Health Rules

```
# /etc/logzilla/forwarder.d/cisco-network-health.yaml
window_size: 0
type: sec
sec_name: cisco-network-health
match:
  - field: cisco_mnemonic
    value:
      - "BGP-5-ADJCHANGE"
      - "BGP-4-MAXPFX"
      - "BGP-3-NOTIFICATION"
      - "OSPF-5-ADJCHG"
rules:
- rewrite:
    program: sec-cisco-network-health
    message: >-
      [CISCO_NETWORK_HEALTH]
      cisco_mnemonic="${:cisco_mnemonic}"
      host="${:host}"
      srcip="${:SrcIP}"
      dstip="${:DstIP}"
      srcport="${:SrcPort}"
      dstport="${:DstPort}"
      protocol="${:protocol}"
      $MESSAGE
```

SEC Rule: BGP Outage Duration Calculation

```
# BGP adjacency outage tracker (host + neighbor)

type=Pair

# ---- START (neighbor Down) ----
ptype=RegExp
pattern=\[CISCO_NETWORK_HEALTH\]\s+cisco_mnemonic="BGP-5-ADJCHANGE"\s+host="([^"]+)"\s+srcip="
[^"]*"\s+dstip="[^"]*"\s+srcport="[^"]*"\s+dstport="[^"]*"\s+protocol="[^"]*"\s+.*\bneighbor\s+([0-
9A-Fa-f:.]+)\s+Down\b
context=BGP_ADJ_$1_$2
desc=BGP Neighbor $2 Down on $1
action=eval %router_host ( "$1" ); \
       eval %neighbor_ip ( "$2" ); \
       eval %down_time ( time() ); \
       shellcmd (/usr/bin/host %neighbor_ip > /tmp/bgp-lookup-%neighbor_ip 2>/dev/null); \
       eval %hostname ( readfile("/tmp/bgp-lookup-%neighbor_ip") ); \
       write /var/log/logzilla/sec/network-health.log \
         "BGP_ADJ_DOWN %t router=%router_host neighbor=%neighbor_ip hostname=%hostname"; \
       shellcmd (logger -t SEC-NETWORK-HEALTH \
         "BGP_ADJ_DOWN router=\"%router_host\" neighbor=\"%neighbor_ip\" hostname=\"%hostname\"")

# ---- END (same host + neighbor, Up) ----
ptype2=RegExp
pattern2=\[CISCO_NETWORK_HEALTH\]\s+cisco_mnemonic="BGP-5-ADJCHANGE"\s+host="([^"]+)"\s+srcip="
[^"]*"\s+dstip="[^"]*"\s+srcport="[^"]*"\s+dstport="[^"]*"\s+protocol="[^"]*"\s+.*\bneighbor\s+([0-
9A-Fa-f:.]+)\s+Up\b
context2=BGP_ADJ_$1_$2
desc2=BGP Neighbor $2 Up on $1
action2=eval %up_time ( time() ); \
        eval %outage_duration ( %up_time - %down_time ); \
        write /var/log/logzilla/sec/network-health.log \
          "BGP_ADJ_UP %t router=%router_host neighbor=%neighbor_ip downtime=%outage_duration
hostname=%hostname"; \
        shellcmd (logger -t SEC-NETWORK-HEALTH \
          "BGP_OUTAGE_RESOLVED router=\"%router_host\" neighbor=\"%neighbor_ip\"
downtime=\"%outage_duration\" hostname=\"%hostname\""); \
        delete /tmp/bgp-lookup-%neighbor_ip

# Pairing window (how long to wait for the Up after Down)
window=86400
```

LogZilla Trigger: BGP Outage Response

```
name: "BGP Outage Business Impact"
filter:
  - field: program
    op: eq
    value: sec-cisco-network-health
```

```
  - field: message
    op: "=~"
    value: "BGP_OUTAGE_RESOLVED"
actions:
  exec_script: true
  script_path: "/usr/local/bin/bgp-outage-analysis.sh"
  send_webhook: true
  send_webhook_template: |
    {
      "event_type": "bgp_outage_resolved",
      "host": "{{:host}}",
      "IP": "{{:SrcIP}}",
      "severity": "warning"
    }
```

Business Intelligence Script

For this script, the goal is to derive data from the log message that is not present in the message itself, in order to send that information to the script. This example uses the LogZilla **Event Enrichment** appstore app.

That app allows administrators to associate additional data with each log message, such as the store type and tunnel ID, based on the IP address of the device in question.

Detailed information about Event Enrichment is available in the **Event Enrichment** appstore entry, which is listed under *Settings*, *App store*, then *Add*. The *Event Enrichment* app icon displays the documentation for the app and how to use it.

For this script Event Enrichment uses **SrcIP** to look up the store name, store number, store type, and tunnel ID.

In `/etc/logzilla/apps/event_enrichment/config/config.yaml` add the following:

```
---
- name: Simple Host Lookup
  description: |
    Used to add store name, store number, store type, and tunnel ID.
    Referenced by SrcIP (the router address for that location).
  metadata_file: StoreInformation.yaml
  lookup_field: SrcIP
```

Next, in `/etc/logzilla/apps/event_enrichment/config/StoreInformation.yaml` add the following:

```
---
"246.219.157.165":
  Store Name: Store AAA
  Store Number: 1001
  Tunnel ID: 123
"107.122.210.185":
  Store Name: Store BBB
```

```
  Store Number: 1002
  Tunnel ID: 234
"CE5-G":
  Store Name: Store CDD
  Store Number: 1003
  Tunnel ID: 345
"PE4":
  Store Name: Store DDD
  Store Number: 1004
  Tunnel ID: 456
```

With the *Event Enrichment* app running using that information, the information will be available to the LogZilla forwarder, and thus to SEC, as user tags.

The forwarder configuration is as follows:

```
# /etc/logzilla/forwarder.d/cisco-network-health.yaml
window_size: 0
type: sec
sec_name: store-network-event
match:
  - field: cisco_mnemonic
    value:
      - "BGP-5-ADJCHANGE"
      - "BGP-4-MAXPFX"
      - "BGP-3-NOTIFICATION"
      - "OSPF-5-ADJCHG"
rules:
- rewrite:
    message: >-
      [STORE_NETWORK_EVENT]
      cisco_mnemonic="${:cisco_mnemonic}"
      host="${:host}"
      srcip="${:SrcIP}"
      dstip="${:DstIP}"
      srcport="${:SrcPort}"
      dstport="${:DstPort}"
      protocol="${:protocol}"
      store_name="${:Store Name}"
      store_number="${:Store Number}"
      tunnel_id="${:Tunnel ID}"
      $MESSAGE
```

The SEC rule would be as follows:

```
# ============================================
# BGP adjacency outage tracker (host + neighbor)
# Uses Event Enrichment app to provide store metadata
# ============================================

type=Pair
```

```
# ---- START (neighbor Down) ----
ptype=RegExp
pattern=\[STORE_NETWORK_EVENT\]\s+cisco_mnemonic="BGP-5-ADJCHANGE"\s+host="([^"]+)"\s+srcip="
([^"]*)"\s+dstip="[^"]*"\s+srcport="[^"]*"\s+dstport="[^"]*"\s+protocol="[^"]*"\s+store_name="
([^"]*)"\s+store_number="([^"]*)"\s+tunnel_id="([^"]*)"\s+.*\bneighbor\s+([0-9A-Fa-f:.]+)\s+Down\b
context=BGP_ADJ_$1_$6
desc=BGP Neighbor $6 Down on $1
action=eval %router_host ( "$1" ); \
       eval %store_ip ( "$2" ); \
       eval %store_name ( "$3" ); \
       eval %store_number ( "$4" ); \
       eval %tunnel_id ( "$5" ); \
       eval %neighbor_ip ( "$6" ); \
       eval %down_time ( time() ); \
       write /var/log/logzilla/sec/network-health.log \
         "BGP_ADJ_DOWN %t router=%router_host neighbor=%neighbor_ip store_ip=%store_ip
store_name=%store_name store_number=%store_number tunnel_id=%tunnel_id"; \
       shellcmd (logger -t SEC-NETWORK-HEALTH \
         "BGP_ADJ_DOWN router=\"%router_host\" neighbor=\"%neighbor_ip\" store_ip=\"%store_ip\"
store_name=\"%store_name\" store_number=\"%store_number\" tunnel_id=\"%tunnel_id\"")

# ---- END (same host + neighbor, Up) ----
ptype2=RegExp
pattern2=\[STORE_NETWORK_EVENT\]\s+cisco_mnemonic="BGP-5-ADJCHANGE"\s+host="([^"]+)"\s+srcip="
([^"]*)"\s+dstip="[^"]*"\s+srcport="[^"]*"\s+dstport="[^"]*"\s+protocol="[^"]*"\s+store_name="
([^"]*)"\s+store_number="([^"]*)"\s+tunnel_id="([^"]*)"\s+.*\bneighbor\s+([0-9A-Fa-f:.]+)\s+Up\b
context2=BGP_ADJ_$1_$6
desc2=BGP Neighbor $6 Up on $1
action2=eval %up_time ( time() ); \
        eval %outage_duration ( %up_time - %down_time ); \
        write /var/log/logzilla/sec/network-health.log \
          "BGP_ADJ_UP %t router=%router_host neighbor=%neighbor_ip downtime=%outage_duration
store_name=%store_name store_number=%store_number tunnel_id=%tunnel_id"; \
        shellcmd (logger -t SEC-NETWORK-HEALTH \
          "BGP_OUTAGE_RESOLVED router=\"%router_host\" neighbor=\"%neighbor_ip\"
downtime=\"%outage_duration\" store_name=\"%store_name\" store_number=\"%store_number\"
tunnel_id=\"%tunnel_id\""); \
        shellcmd (/usr/local/bin/bgp-outage-analysis.sh "%outage_duration" "%store_name"
"%store_number" "%tunnel_id")

window=86400
```

```
#!/bin/bash
# /usr/local/bin/bgp-outage-analysis.sh

DOWNTIME=$1
STORE_NAME=$2
STORE_NUMBER=$3
TUNNEL_ID=$4

# Validate inputs
if [ -z "$DOWNTIME" ] || [ -z "$STORE_NUMBER" ] || [ -z "$TUNNEL_ID" ]; then
```

```
    logger -t BGP-BUSINESS "ERROR: Missing required parameters"
    exit 1
fi

# Convert seconds to human readable
HOURS=$((DOWNTIME / 3600))
MINUTES=$(((DOWNTIME % 3600) / 60))
DURATION_TEXT="${HOURS}h ${MINUTES}m"
BUSINESS_IMPACT="standard"

# Tunnel 241 gets special handling (critical business locations)
if [[ "$TUNNEL_ID" == "241" ]]; then
    BUSINESS_IMPACT="high"

    # Alert management for critical tunnel outages > 4 hours
    if [[ "$DOWNTIME" -gt 14400 ]]; then
        curl -s -X POST "https://slack.company.com/api/webhooks/management" \
             -H "Content-Type: application/json" \
             -d "{\"text\":\"CRITICAL: Store $STORE_NUMBER (Tunnel 241) was down for
$DURATION_TEXT\"}" \
             2>/dev/null || logger -t BGP-BUSINESS "ERROR: Failed to send Slack alert"
    fi
fi

# Log business metrics
logger -t BGP-BUSINESS "Store: $STORE_NUMBER, Name: $STORE_NAME, Duration: $DURATION_TEXT, Impact:
$BUSINESS_IMPACT, Tunnel: $TUNNEL_ID"

# Update network operations dashboard
curl -s -X POST "https://dashboard.company.com/api/bgp-events" \
     -H "Content-Type: application/x-www-form-urlencoded" \
     -d
"store=$STORE_NUMBER&store_name=$STORE_NAME&duration=$DOWNTIME&impact=$BUSINESS_IMPACT&tunnel=$TUNNEL
_ID" \
     2>/dev/null || logger -t BGP-BUSINESS "ERROR: Failed to update dashboard"

exit 0
```

BGP Flapping Detection

Multi-Event BGP State Correlation

Detect BGP neighbors that change state multiple times within a short period.

SEC Rule: BGP Flapping Detection

```
# Track BGP-5-ADJCHANGE state changes per neighbor
type=SingleWithThreshold
```

```
ptype=RegExp
pattern=\[CISCO_NETWORK_HEALTH\]\s+cisco_mnemonic="BGP-5-ADJCHANGE"\s+host="
([^"]+)"\s+.*\bneighbor\s+([0-9A-Fa-f:.]+)\s+(Up|Down)
context=BGP_FLAP_$2
desc=BGP neighbor flapping detected
action=eval %hostname ( "$1" ); \
       eval %neighbor ( "$2" ); \
       eval %state ( "$3" ); \
       create BGP_FLAP_%neighbor 3600; \
       write /var/log/logzilla/sec/network-health.log \
         "BGP_FLAP_DETECTED %t neighbor=%neighbor host=%hostname state=%state flap_count=$thresh
window=600s"; \
       shellcmd (logger -t SEC-NETWORK-HEALTH \
         "BGP_FLAP_DETECTED neighbor=\"%neighbor\" host=\"%hostname\" state=\"%state\"
flap_count=\"$thresh\" window=\"600s\"")
thresh=5
window=600
```

BGP Route Advertisement Monitoring

Route Withdrawal/Advertisement Correlation

Monitor BGP route advertisements and detect routing instability.

SEC Rule: Route Instability Detection

```
# Route Instability Detection
# Detects excessive BGP route-related events per neighbor

type=SingleWithThreshold

ptype=RegExp
pattern=\[CISCO_NETWORK_HEALTH\]\s+cisco_mnemonic="BGP-[34]-(?:MAXPFX|NOTIFICATION)"\s+host="
([^"]+)"\s+srcip="([^"]+)"\s+dstip="[^"]*"\s+srcport="[^"]*"\s+dstport="[^"]*"\s+protocol="[^"]*"\s+
(.*)

context=ROUTE_INSTABILITY_%neighbor_ip
desc=Route instability from BGP neighbor $2 on $1
action=eval %router_host ( "$1" ); \
       eval %neighbor_ip ( "$2" ); \
       eval %message_detail ( "$3" ); \
       eval %route_prefix ( (%message_detail =~ /Prefix\s+([\d.\/]+)/) ? "$1" : "N/A" ); \
       eval %addr_family ( (%message_detail =~ /\((IPv[46]|VPNv[46])[^)]*\)/) ? "$1" : "N/A" ); \
       eval %vrf ( (%message_detail =~ /VRF\s+(\S+)/) ? "$1" : "default" ); \
       create ROUTE_INSTABILITY_%neighbor_ip 3600; \
       write /var/log/logzilla/sec/network-health.log \
         "ROUTE_INSTABILITY %t router=%router_host neighbor=%neighbor_ip prefix=%route_prefix
addr_family=%addr_family vrf=%vrf event_count=$thresh window=300s"; \
```

```
        shellcmd (logger -t SEC-NETWORK-HEALTH -p local0.warning \
          "ROUTE_INSTABILITY router=\"%router_host\" neighbor=\"%neighbor_ip\"
prefix=\"%route_prefix\" addr_family=\"%addr_family\" vrf=\"%vrf\" event_count=\"$thresh\"")

thresh=10
window=300
```

Multi-Router BGP Correlation

Network-Wide BGP Event Correlation

Detect BGP events affecting multiple routers simultaneously, indicating upstream provider issues.

SEC Rule: Network-Wide BGP Correlation

```
# Multi-Router BGP Impact Detection
# Detects when same BGP neighbor affects 3+ routers within 5 minutes

type=SingleWithThreshold

ptype=RegExp
pattern=\[CISCO_NETWORK_HEALTH\]\s+cisco_mnemonic="BGP-[345]-
(?:ADJCHANGE|NOTIFICATION|MAXPFX)"\s+host="([^"]+)"\s+srcip="([^"]+)"\s+dstip="[^"]*"\s+srcport="
[^"]*"\s+dstport="[^"]*"\s+protocol="[^"]*"\s+

context=BGP_MULTI_ROUTER_IMPACT_%neighbor_ip_HOST_%router_host
desc=BGP events from neighbor $2 affecting multiple routers
action=eval %router_host ( "$1" ); \
       eval %neighbor_ip ( "$2" ); \
       create BGP_NEIGHBOR_IMPACT_%neighbor_ip 300 \
         (report BGP_NEIGHBOR_IMPACT_%neighbor_ip \
           /bin/bash -c 'echo "%neighbor_ip affecting multiple routers" | \
             logger -t SEC-NETWORK-HEALTH -p local0.crit \
               "BGP_NETWORK_OUTAGE neighbor=\"%neighbor_ip\" affected_routers=\"$thresh\"
window=\"300s\" - possible upstream provider issue"'); \
       add BGP_NEIGHBOR_IMPACT_%neighbor_ip %router_host; \
       write /var/log/logzilla/sec/network-health.log \
         "BGP_NETWORK_OUTAGE %t neighbor=%neighbor_ip router=%router_host total_routers=$thresh"

thresh=3
window=300
```

LogZilla Trigger: Network-Wide BGP Outage Response

```
name: "Network-Wide BGP Outage"
filter:
  - field: program
    op: eq
    value: SEC-NETWORK-HEALTH
  - field: message
    op: "=~"
    value: "BGP_NETWORK_OUTAGE"
actions:
  send_email: true
  send_email_template: |
    Subject: CRITICAL: Network-Wide BGP Outage

    Network-wide BGP outage detected affecting multiple routers.

    Host: {{:host}}
    Message: {{:message}}

    This indicates a potential upstream provider issue.
    Escalate to network operations immediately.
  issue_notification: true
```

BGP Security Monitoring

BGP Hijack Detection

Monitor for unexpected BGP route advertisements that could indicate hijacking.

SEC Rule: BGP Hijack Detection

```
# BGP Route Anomaly Detection
# Detects suspicious BGP activity via malformed updates and prefix flooding

type=Single

ptype=RegExp
pattern=\[CISCO_NETWORK_HEALTH\]\s+cisco_mnemonic="BGP-[34]-(NOTIFICATION|MAXPFX)"\s+host="
([^"]+)"\s+srcip="([^"]+)"\s+dstip="[^"]*"\s+srcport="[^"]*"\s+dstport="[^"]*"\s+protocol="[^"]*"\s+
(.*)

desc=Suspicious BGP activity detected from neighbor $3 on $2
action=eval %event_type ( "$1" ); \
       eval %router_host ( "$2" ); \
       eval %neighbor_ip ( "$3" ); \
       eval %message_detail ( "$4" ); \
```

```
        eval %error_code ( (%message_detail =~ /(\d+\/\d+)/) ? "$1" : "N/A" ); \
        eval %error_desc ( (%message_detail =~ /\(([^)]+)\)/) ? "$1" : "N/A" ); \
        write /var/log/logzilla/sec/network-health.log \
          "BGP_ANOMALY_DETECTED %t event=%event_type router=%router_host neighbor=%neighbor_ip
error_code=%error_code error_desc=%error_desc"; \
        shellcmd (logger -t SEC-SECURITY -p local0.warning \
          "BGP_ANOMALY_DETECTED event=\"%event_type\" router=\"%router_host\"
neighbor=\"%neighbor_ip\" error=\"%error_desc\" - REVIEW REQUIRED")
```

References

- Brute Force Attack Detection (https://www.logzilla.ai/docs/event-correlation/brute-force-attack-detection)
- Network Infrastructure Correlation (https://www.logzilla.ai/docs/event-correlation/network-infrastructure-correlation)
- Advanced Event Correlation Walkthrough (https://www.logzilla.ai/docs/event-correlation/advanced-event-correlation-walkthrough)