Tamper detection

The Tamper Detection feature is designed to enhance the physical security by identifying unauthorized access to the device's internal components. Each device is shipped from the factory with its cover screwed tightly. If the cover is subsequently opened, the hardware records these open or close events in a tamper-proof memory. This recording occurs whether the system is powered on or off. Upon boot-up, the IOS software reads these recorded events and compares them against previously known event indices. If discrepancies are found, the software generates a syslog message, alerting the user to a potential tamper event.

Note The ability to send the alarm depends on the the WAN interface (Ethernet or LTE connectivity) being available at the time of the occurrence of the tamper detection.

Tamper detection capabilities and event logging

The tamper detection hardware operates independently of the software configuration, continuously recording cover open or close events. These events are stored in two dedicated memory sections:

1. **System fully powered off**:
   - When the system is powered off, a battery powers the hardware to record events.
   - Only the *first* cover open event between the last power-off and the next power-on is recorded, even if multiple open/close events occur during this period.
   - This section can log up to 250 events (2KB). If the log capacity is reached, it will roll over, and a rollover flag will be set.

2. **System partially or fully powered up**:
   - When system power is available, the hardware records all cover open or close events.
   - This section can log up to 506 events (4KB). Similar to the power-off section, the log will roll over if full, and a rollover flag will be set.

Restrictions of tamper detection

- During a fully powered-off state, only the *first* cover open event is recorded. Subsequent open or close events during the same power-off cycle are not logged.
- Each event log section has a finite capacity (250 for power-off, 506 for power-on). Once full, the logs roll over, potentially overwriting older events, though a rollover flag indicates this.

How to configure tamper detection

Tamper detection is enabled by default, meaning the hardware continuously records events. The software configuration primarily controls whether syslog notifications are displayed during boot-up when new tamper events are detected.

To prevent the system from displaying syslog messages related to tamper events during boot-up, you can disable the feature's notification capability. This does not stop the hardware from recording events.

Router(config)# no platform tamper-detection

When this command is configured, users will not see the tamper event messages during boot-up, even if new events have occurred.

Procedure

Step 1 request consent-token generate-challenge tamper-auth auth-timeout <timeout value in minutes>

Generates a challenge for consent token.

Step 2 request consent-token accept-response tamper-auth <response_string>

Accept the response with the generated challenge.

Step 3 request platform hardware tamper-detection event-mark

Post successful execution, the system confirms that the tamper event has been marked and terminate the authentication session.

Examples

Dumping event logs

A `show` command is available to dump all or a specified number of recent tamper events from either the power-on or power-off log sections. This command is accessible regardless of whether tamper detection notifications are enabled or disabled.

Example 1: Showing the last 4 power-on tamper events

```
Router# show platform tamper-detection event power-on lastx 4

Current Time: 2025/02/11 08:22:07    Rollover Status: No    Rollover Count: 0
------------------------------------------------------------------------------------------
Tamper event index   |   Tamper event timestamp   |   Tamper events description
------------------------------------------------------------------------------------------

    #6              2023/08/01 10:33:21          Chassis is opened
    #5              2023/08/01 10:33:15          Chassis is closed
    #4              2023/08/01 10:33:15          Chassis is opened
    #3              2023/08/01 10:33:15          Chassis is closed
```

The output includes the event index, timestamp, and a description of the event (Chassis is opened/closed).

Example 2: Showing power-on events with log rollover

```
Router# show platform tamper-detection event power-on lastx 2

Current Time: 2023/10/01 15:28:56    Rollover Status: Yes    Rollover Count: 5
------------------------------------------------------------------------------------------
Tamper event index   |   Tamper event timestamp   |   Tamper events description
------------------------------------------------------------------------------------------

    #2627           2023/08/12 08:44:40          Chassis is closed
    #2626           2023/08/12 08:41:27          Chassis is opened
```

In this example, "Rollover Status: Yes" and "Rollover Count: 5" indicate that the event log has rolled over 5 times, meaning older events have been overwritten.

Consent token authentication example

The `event-mark` command requires consent token authentication for security. Below is an example of the workflow:

```
Router# request consent-token generate-challenge tamper-auth auth-timeout 60
[illegible]
Router#
*Feb 11 08:48:39.682: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (challenge generation
attempt: Tamper-auth 0).

8PP2B# request consent-token accept-response tamper-auth
61+O/gAAAQYBAAQAAAUBAgAEAAAAAAMBYlNVOHNvRmlIUDFiUEp4OU5yTVEwUCtK
SkMrNUp3MnpNMWlJcC9ibEJjakQxcFJaWGxtNDNkWGJVYnZ2bVpiRFUNCjV2ckt3
eDV4ajEabnVtUURNTFdxeExLb0IwMHhkMkRRdEQyUWJ2TlR5L0V4V1JNRjdDQ0VT
YjRKREFrVnBaMGkNCm43ZHlyMWsrTGpGV1FaeGkvc2lnUVgyOHR4ZEx6VHVsckxk
b3pGK3FnRmZCeWNhR3lmU3VwT0ZKcFU2T0ZGVWANCllhY01ZWXZ3aTF0WUZKaE1n
d1BBTVkxbFE4b3JoSXVNVTU5S1RhZ0YxNmw0V0tBK3dEbVpHc2F0Sk5ubDIrL3gN
CjdndmljMFNCNXR2aXR2YVNyVTNtaVFnWkM3MnNLdHhMb3R5Zi9nNVVwcWVkUFJR
K21aVXhlUysyakdFS1pNbmoNCnIrNkpIQnpXZ05lMm05SUpnUGhTVHc9PQoBYk9O
NCtGSi95Ym1LeVpHRmNBY28xQjZlUVBrcSswY3lWU0lsZkgyVGw1dFNHM0NjWHk4
bnpCa0xIYTN2NGZJMjQNClZ5aE96MldTZDRBMUk1bHFVVmtyekI3aHJqRjJsaUQ1
anRQOU1XcEVaMkR3eVMxcmh0bmhHNG1HRFFiQmFXVlENCmJXbVBQUkFiQ2xobG1a
L2RDUTdhMkxIVXA0RDI0RTJPWmtsdGlXeVpnT3dBVHZQUndXVWxmTm9HSlNxN3U4
a0QNClZBM2pqTW5MdHJXRzFkTkhZemg4UjM3Uy82SDVXY1dJOWpZMGlkS2NqRDVC
TzUzK21xYTdLNVU2T3ppaE5FZFcNCjQvRlk1OStldWs2d0kyeDZTdWE1V2dZKy8w
UnIzL3B4RTJjTDg5NUVvejYwWlNnaFc4d25LYU1RMEM0cWFlc1kNCmNYanc3alZm
ekpZY0dkTkpXK280a0E9PQ==

% Consent token authorization success

Router#
*Feb 11 08:49:29.826: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (authentication success:
 Tamper-auth 0).
```

```
Router# request platform hardware tamper-detection event-mark
% Tamper-event marked successfully...terminating the tamper auth session
% Consent token authorization termination success

8PP2B#
*Feb 11 08:49:54.788: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (terminate authentication:
 Tamper-auth 0).
```