OPEN Compute Summit

January 28–29, 2014 San Jose

“IPMI over HTTPS”

Overview of Intel’s presentation/proposal to the IPMI Promoters Group

OPEN
Compute Summit

January 28–29, 2014 San Jose

Tom Slaight

Intel

Principal Server Manageability Architect

Disclaimers

This presentation contains material that has been proposed by Intel to the IPMI Promoters Group.

This information is subject to change without notice. Presentation of this material does not imply any commitment to its inclusion in the IPMI specifications, nor does it imply any endorsements by the IPMI Promoters, nor by Intel on behalf of the IPMI Promoters or the IPMI Promoters Group.

Looking for general feedback / interest level on the concept

Background

- The present IPMI Specification uses a transport protocol called “RMCP+” (Remote Management Control Protocol +)
    - Introduced with IPMI v2.0 in February, 2004.
    - Designed to fit on 8-bit microcontrollers with limited resources and network connectivity.
        - UDP based. Low processing overhead for IPMI messages.
    - Uses standard algorithms for integrity and confidentiality
        - Supports HMAC-SHA256 and AES-128
    - Session establishment uses a key exchange protocol called “RAKP” (Remote Access Key exchange Protocol)
        - Form of symmetric challenge/response with pre-shared keys.
    - 20 byte (160 bit) ‘per user’ key + additional 20 byte (160 bit) “Kg” ‘per bmc’ key.

Why IPMI over HTTPS ?

- Increased user desire well-known “Internet Standard” protocols

- Easier to develop remote applications
  - More development tools available for HTTPs/TLS
  - Greater developer familiarity with HTTP formats

- Takes advantage of TLS/HTTPS infrastructure
  - For session integrity and confidentiality

- BMC Performance is less of an issue
  - 8- and 16-bit BMCs have migrated to 32-bit
  - Execution speed has increased
  - Many BMCs still have RAM limitations

Supports standard transport while preserving IPMI’s
‘byte efficiency’ and ecosystem

Proposal Highlights

- Uses TLS with certificates for the remote-console to BMC connection
  - TLS is used to provide the encryption (confidentiality) and integrity for the connection.
    - Including protection from low-level man-in-middle and replay attacks.

- Uses HTTP and JSON as the transport protocol for IPMI operations
  - DISCOVER OPERATION SUPPORT
  - GET SESSION CHALLENGE
  - IPMI USER LOGIN
  - IPMI USER LOGOUT (also terminates HTTPS session)
  - TRANSFER IPMI MESSAGE
  - TRANSFER SOL DATA
  - TRANSFER OEM/ORG DATA

- IPMI LAN Alerting is unchanged
  - Continues to use SNMP-based “Platform Event Trap” format

Next Level

- HTTP PUTs only
  - Good fit with IPMI messaging request/response approach
  - Simplifies implementation and parsing
- Uses IPMB-based Message Format
  - Leverages existing tools (just replaces RMCP+ transport layer)
  - Simplifies mixed support of HTTPS and RMCP+ BMCs at same site
- Includes IPMI-specified User Login exchange
  - Provides a basic BMC attestation mechanism
  - Assumes TLS Certificates themselves may not be authorized or trusted and consequently that BMCs may be spoofed.
- Does not introduce a ‘human readable’ interface for IPMI
- OEM Payload support is not part of proposal
  - Can use OEM/ORG –specific transfers instead

Example: IPMI Message Transfer

```
PUT /IPMI/msg/session_ID HTTP/1.1
Host: example.org
User-Agent: IPMI/1.0
Content-length: nnn
Content-type: application/json
{
   "DATA": "msg_data",
   "GET": "y_n"
}

HTTP/1.1 200 Ok
Date: Wed, 20 Dec 2013 17:02:12 GMT
Content-Length: nnn
Content-type: application/json
{
   "STATUS": "status",
   "DATA_R": "r_msg_data",
   "GET_R": "r_y_n"
}
```

The requested IPMI operation is identified in the URI of the PUT

msg_data = IPMI message data in hex-ascii encoding. The transfer is indifferent to whether the msg_data holds a request or a response message.

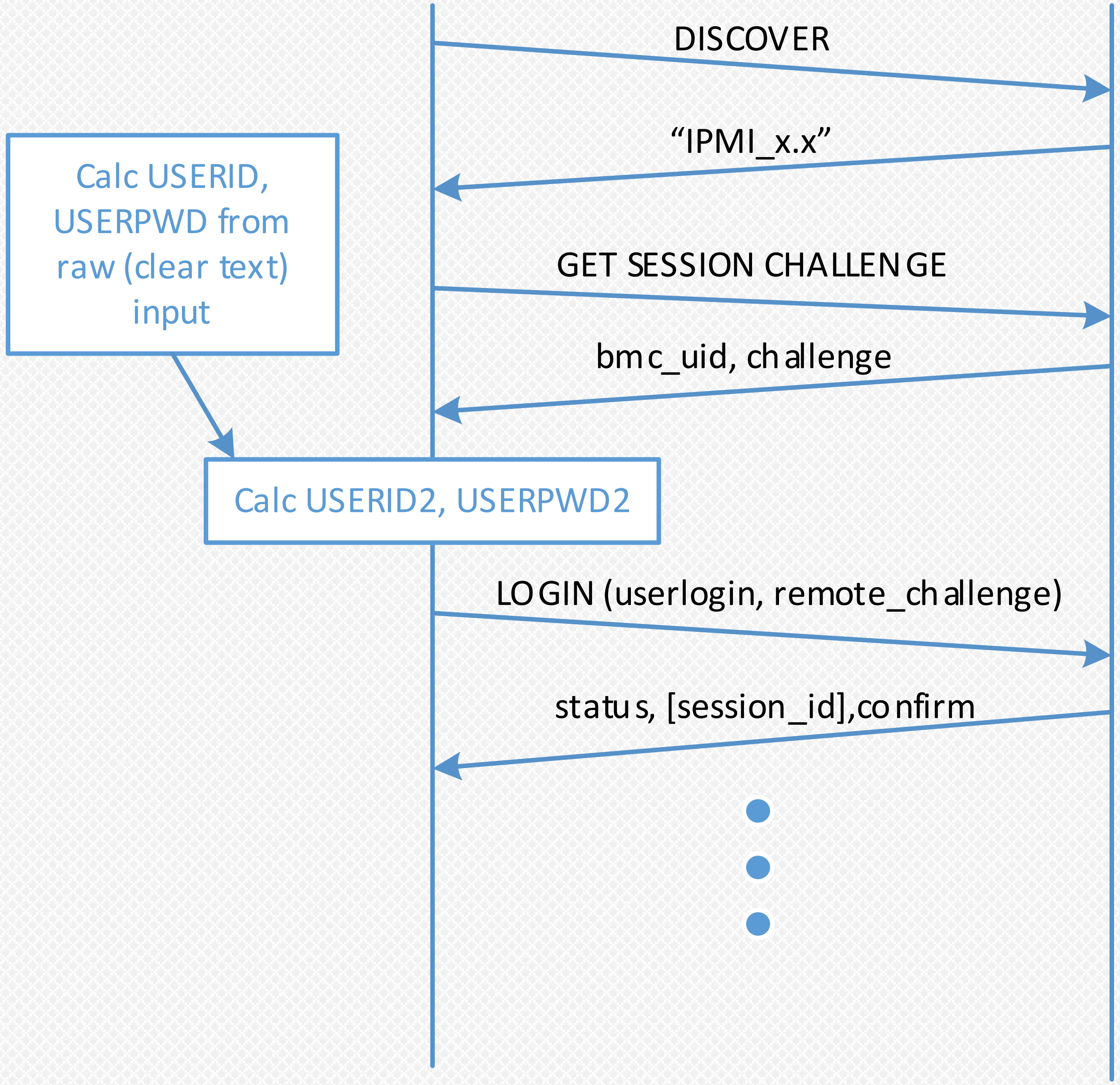
The “GET” parameter indicates whether the remote console wishes to get any outgoing message from the BMC. The remote console can choose to only transmit msg_data, only request r_msg_data, or do both.

The status code indicates status for the IPMI-over-https transfer only. It does not replace the IPMI Completion Code.

E.g. “OK” means the BMC accepted the requested operation from the PUT, but does not indicate whether the msg_data content itself is correct.

Simple definition IPMI Message transfer over an alternative transport

Example: Login flow


DISCOVER
“IPMI_x.x”
Calc USERID, USERPWD from raw (clear text) input
GET SESSION CHALLENGE
bmc_uid, challenge
Calc USERID2, USERPWD2
LOGIN (userlogin, remote_challenge)
status, [session_id],confirm

OPEN

Compute Summit

January 28–29, 2014 San Jose