VOLTTRON^{TM}: Security Features and Discussion

BRANDON CARPENTER, BORA AKYOL

Pacific Northwest National Laboratory

Software Framework for Transactive Energy: VOLTTRON™, VTARI, Arlington, VA

VOLTTRON Team

Software Development Team

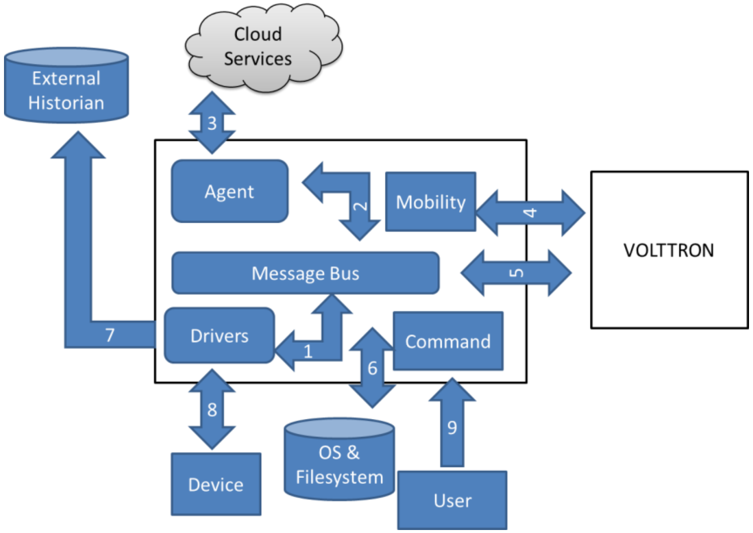
- Bora Akyol
- Jereme Haack
- Brandon Carpenter
- Kyle Monson
- Craig Allwardt
- Poorva Sharma
- Tim Kang
- Robert Lutes
- Casey Neubauer
- Dan Johnson

Application Development Team

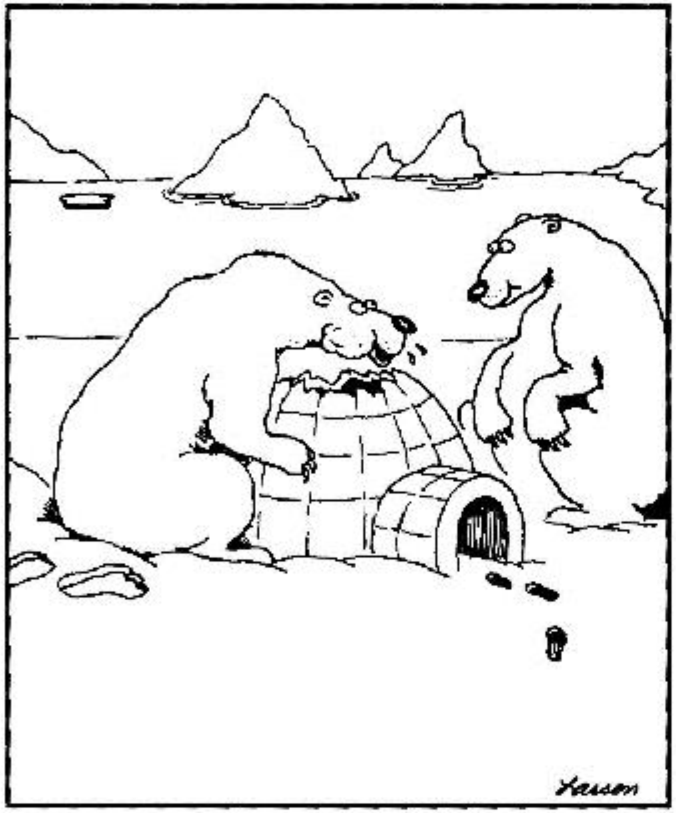
- Srinivas Katipamula
- Robert Lutes
- Wooyun Kim
- Rick Pratt
- Carl Miller
- Weimin Wang
- Siddartha Goyal
- Michael Brambley
- Lucy Huang
- Chad Corbin
- He Hao

Overview

- Threat modeling for context
- List of security features/deployment recommendations
- Recommends review of NIST SP800-82 Guide to Industrial Control Systems (ICS) Security
  - http://csrc.nist.gov/publications/drafts/800-82r2/sp800_82_r2_draft.pdf
- Please make suggestions as github issues or email to volttron@pnnl.gov



Security in VOLTTRON 2.x



"Oh, hey! I just love these things! ... Crunchy on the outside and a chewy center!"

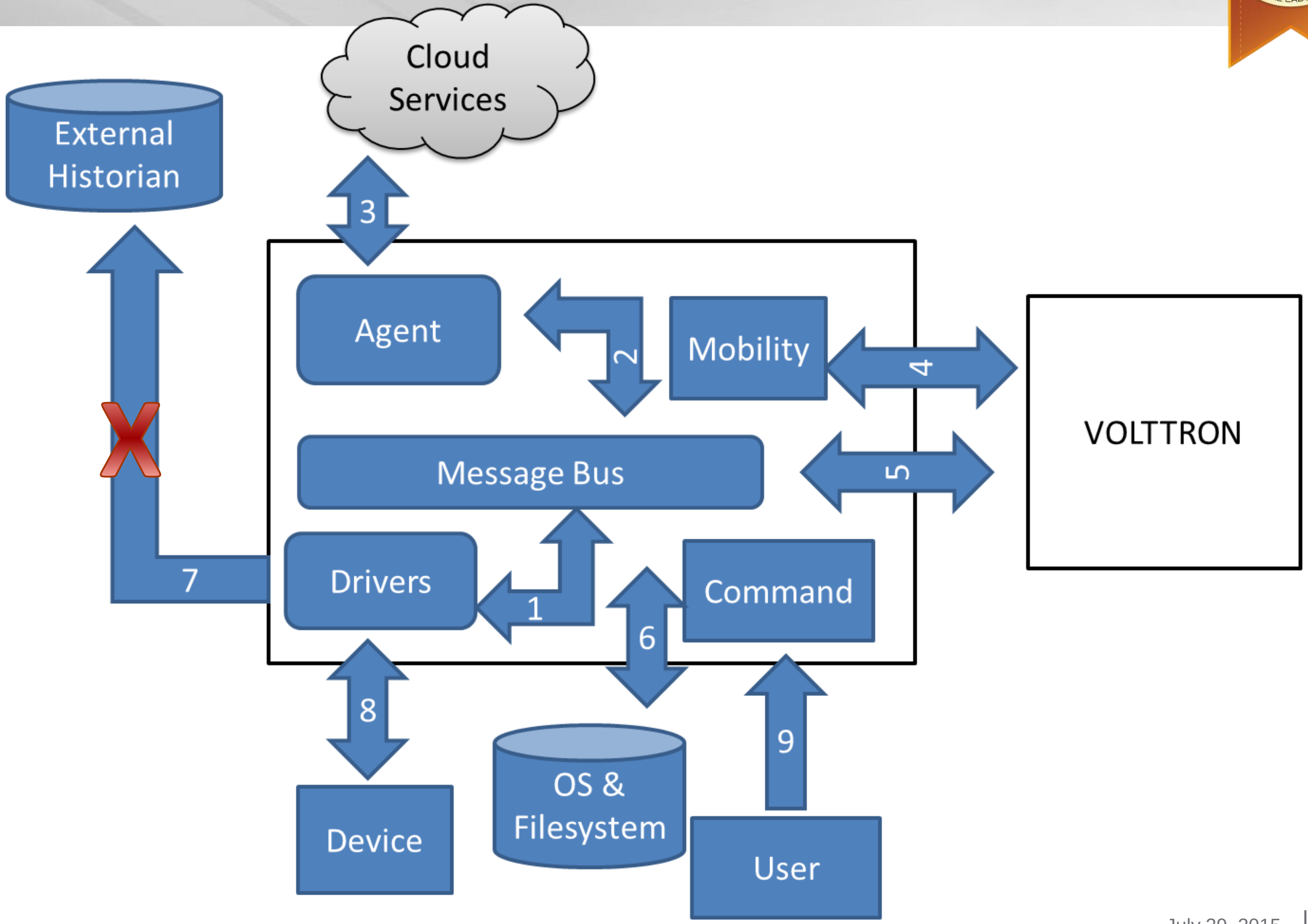
VOLTTRON Security Goals (including 3.0 release)

- Protecting the integrity of agent programming through cryptographic means
- Protecting agents from using excessive system resources to ensure platform stability
- Protecting agent configuration (and work orders) from manipulation
- Securing communications between VOLTTRON platforms and external data sources
- Securing communications between platform instances, including the transfer of agents
- Securing communications between agents running on the same VOLTTRON platform

VOLTTRON 3.0 Security Improvements

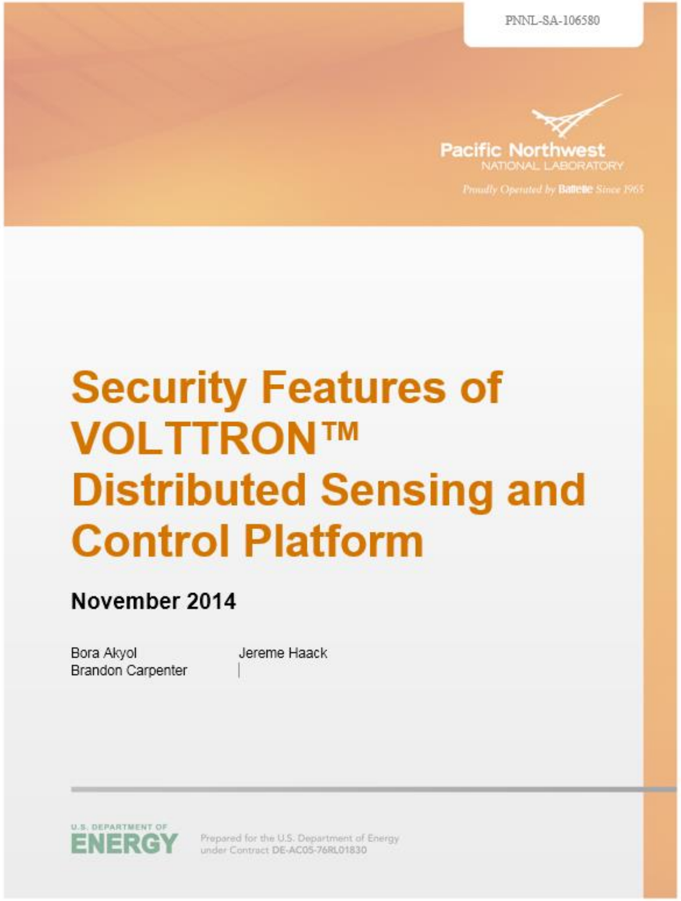
- Platform hardening recommendations for all VOLTTRON users and use cases
- CurveMQ encryption and authentication enabled by default for TCP connections into the platform
- All connections are authenticated
- Authorization based on identity, domain, endpoint, and authentication credentials
  - All messages include user ID for authorization and attribution
  - Associated on the platform and cannot be spoofed by agents
- Platform refuses to run as root to prevent damage and escalation

Components and Communication



Threat Identification and Mitigation Strategies

- VOLTTRON security features and documented in the white paper
- The paper first identifies possible attack vectors against the VOLTTRON software, as well as associated risks and mitigations for reducing, alleviating, or even eliminating the risks
- Some threats also include future strategies for improving the mitigations and even further reducing the risk
- Each vulnerability follows the following template:
  - Description of vulnerability/threat.
  - **R**: Associated risk indicating what might occur if the vulnerability is exercised.
  - **M**: Mitigation that should/will be implemented to reduce or eliminate the associated risk.
  - **F**: Future strategies for mitigation (optional)



PNNL-SA-106580

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Security Features of VOLTTRON™ Distributed Sensing and Control Platform

November 2014

Bora Akyol
Brandon Carpenter

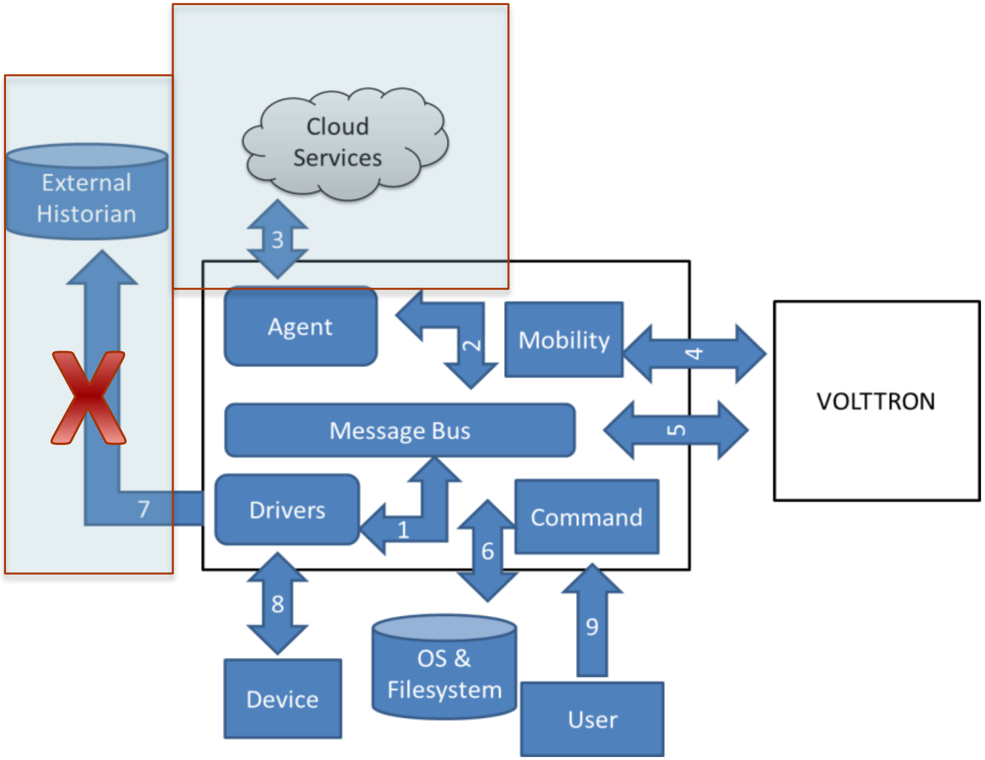
Jereme Haack

U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Communications between VOLTTRON and other services (3 and 7)

- VOLTTRON allows agents to act as proxies to external resources to move information between a service and the platform (3).
- VOLTTRON utilizes an external service for data storage
  - Devices managed by the platform
  - ~~Application log messages (7).~~

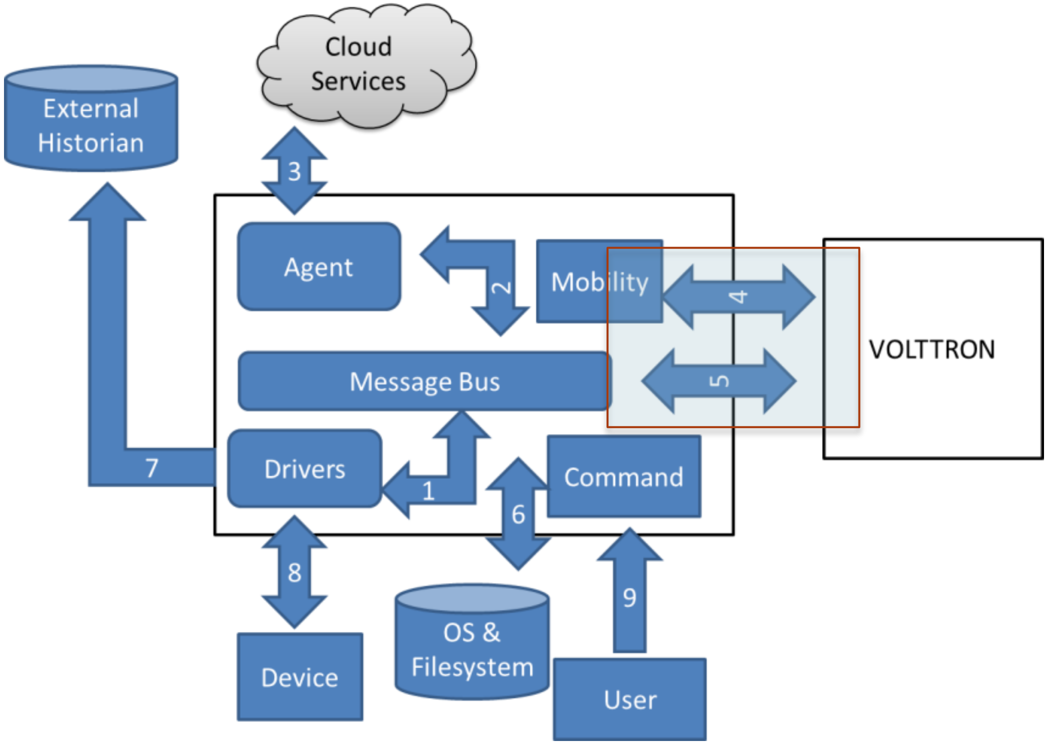


Example Threats and Mitigations

- Communication between agents and the Cloud/external historian could be intercepted, tampered with or snooped upon by a third party
  - **R:** A remote entity can insert itself between VOLTTRON and external entities. Once in the path, the remote entity can tamper with communications, affecting integrity, or snoop the communications, affecting confidentiality.
  - **M**: VOLTTRON uses standardized communication protocols (TLS/SSL) when communicating with external entities. These protocols provide both message integrity and confidentiality services. Identities are authenticated by means of X509 certificates.
- Agents may communicate with any system in the Cloud with which any other agent is also able to communicate (assumes firewall allows such communication).
  - **R:** An agent may exfiltrate data to systems in the Cloud.
  - **M**: Agent code will be reviewed for security issues and malicious intent.
  - **F**: Agents will be sandboxed to disallow unauthorized communications.

Communications between multiple VOLTTRON instances (4 and 5)

- VOLTTRON platforms can communicate with each other through the multi-node communication service (5).
- Mobility Service (4) enables agents to move between platforms and enables administrators to provision VOLTTRON devices in the field.

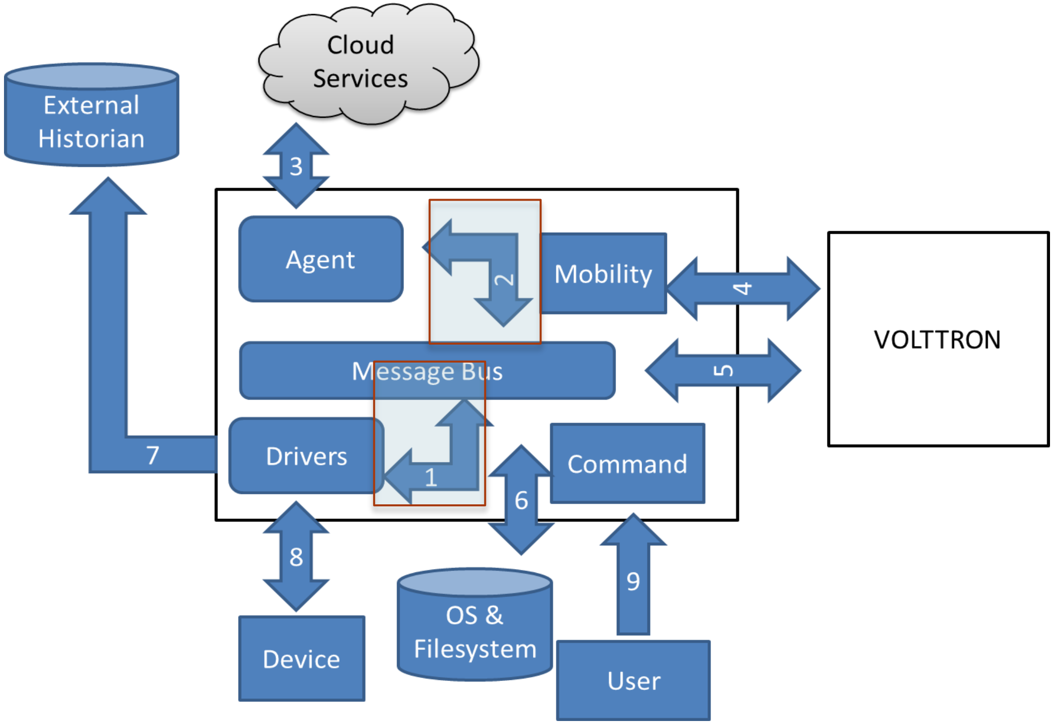


Example Associated Vulnerability

- Agents may communicate between the messaging buses of platforms located on different systems (if the multi-node agent is enabled).
  - **R**: Multi-node messages may cross uncontrolled networks providing opportunity for interception or modification.
  - **M**: Multi-node messaging uses the elliptic-curve encryption technology of ØMQ's CurveMQ protocol to authenticate and encrypt traffic between nodes (when a keyset is provided).
  - **R**: An additional TCP (transmission control protocol) port is required for multi-node communication and mobility service which may be susceptible to denial of service (DoS) attacks.
  - **M**: Firewall rules may be applied to help limit the effectiveness of such attacks. VOLTTRON provides no protection itself against DoS attacks.
  - **M**: Opening a single port and multiplexing traffic limits the number of network ports requiring exposure to the Internet, reducing the attack surface.

Agent multi-tenancy inside a VOLTTRON platform (1,2, and others)

- VOLTTRON supports multiple agents and services to be running simultaneously. This ability is referred to as multi-tenancy. The following are potential vulnerabilities related to agent and service multi-tenancy inside the platform:

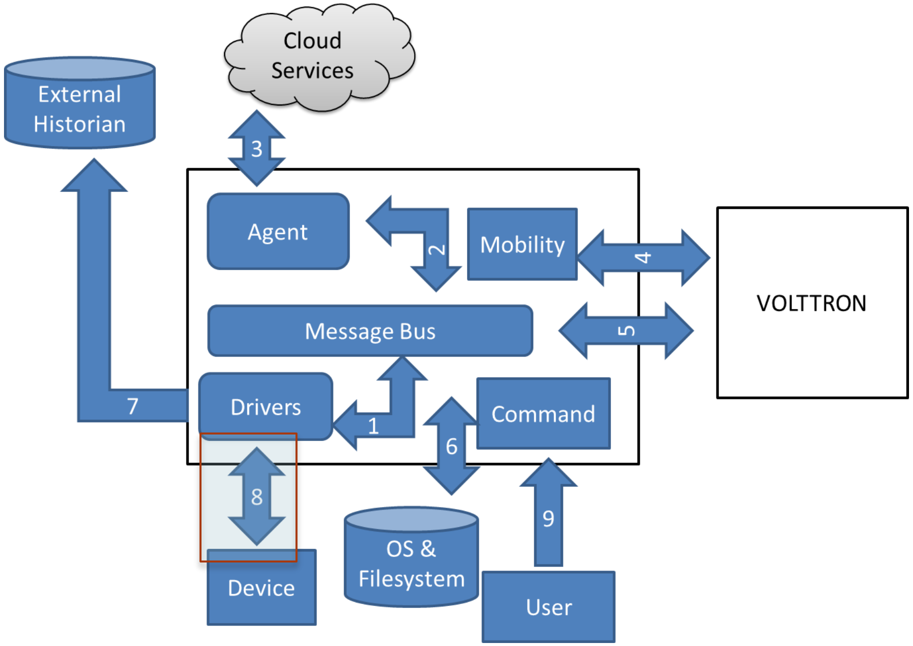


Example Vulnerability and Mitigation

- All agents run under the same user account, with the same privilege level.
  - **R**: A malicious agent can interfere with other agents, possibly sending them signals, killing them, or overwriting data.
  - **M**: Agent code will be reviewed for security issues and malicious intent.
  - **M**: Agent code is signed (multiple times) and verified before each execution. This prevents an unauthorized third party from tampering with agent code.
  - **F**: Full containerization of agent code is planned for a future release and will effectively isolate agents.
- Local communication between agents over the message bus is unencrypted.
  - **R**: A malicious agent may subscribe to every message sent on the message bus and retransmit it or use it for other unintended purposes.
  - **M**: Operating system and VOLTTRON level features defend against compromise of the internal messaging bus.
  - **M**: Private pub/sub and peer-to-peer communications are supported.
  - **F**: Encrypting the body of inter-agent communications may be supported.

Communicating with and controlling devices (8)

- VOLTTRON drivers (8) allow the platform to both collect data from devices and send control commands utilizing protocols such as Modbus, BACnet, or custom-built software. Data is then published to the message bus
  - Overall security of the underlying communications protocol such as BACnet is outside the scope of this document.

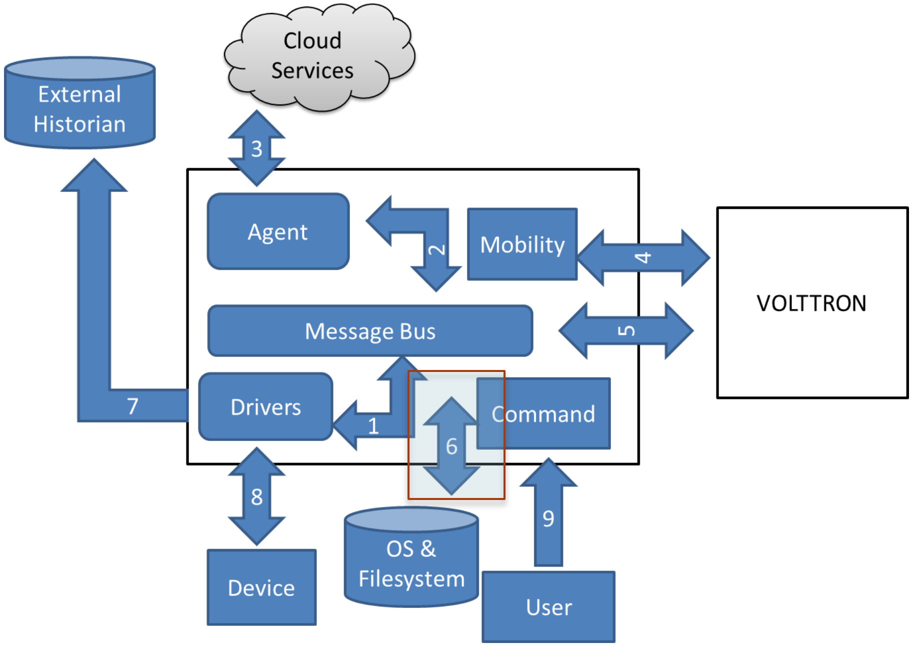


Example Threat and Mitigation

- Unsecured communications between VOLTTRON and legacy control devices (e.g. Modbus, BACnet) may be intercepted and modified by a third party.
  - **R**: A third party can modify communications between VOLTTRON and controlled devices using legacy protocols. It is even possible to impersonate a controlled device. An agent can then react incorrectly to information coming from such a device.
  - **M**: Security measures as described in NIST SP800-82 and Neilson are to be used to protect legacy control system devices that do not have sufficient security protections built-in.
  - **M**: VOLTTRON agents can be written to validate information being received from legacy devices to check for range, historically known trends, etc.
  - **M**: VOLTTRON device drivers are written to prevent potential exploits from “overflow” type attacks to gain access to the platform by means of data being passed by a legacy device. VOLTTRON platform can not be compromised by means of incorrect data streams.

Underlying operating system and file system (Interface 6)

- VOLTTRON is built on top of a modern Linux operating system.
  - Security hardening of the operating system
  - Keeping up-to-date with periodically applied patches to further enhance the security of the instances deployed in the field.

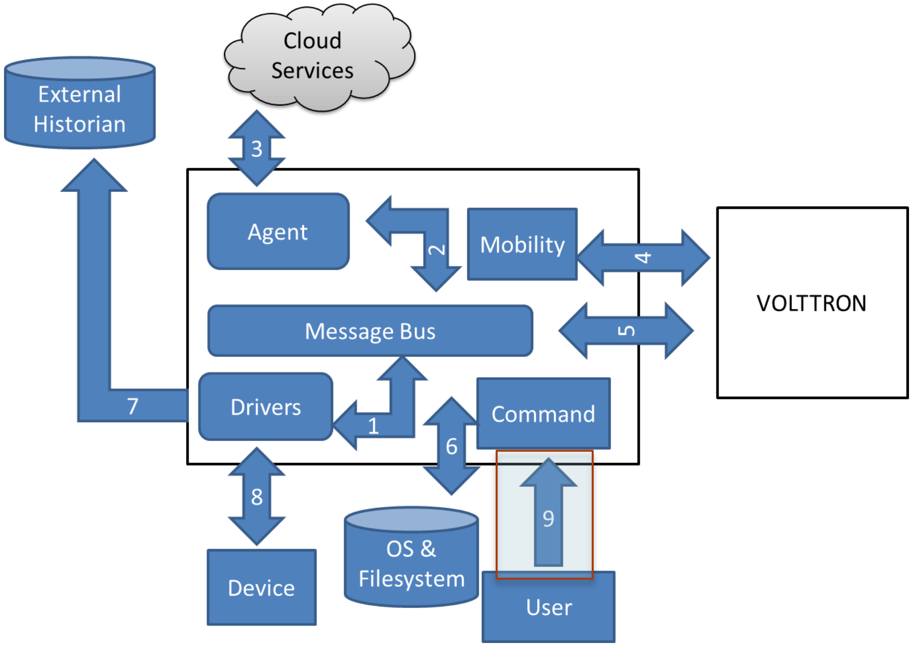


Example Threat and Mitigation

- Platform supervisor and agents run on a shared (multi-user) system (6).
  - **R**: Other users on the system might be able interact with supervisor or agent processes, files, and/or sockets. A malicious agent might be able to access processes, files, and sockets belonging to other users of the system.
  - **M**: In a production environment, the supervisor should run under its own unprivileged account. Files are writable only by the supervisor process's owner. Sensitive files are only readable by the supervisor process's owner. Appropriate file system permissions are set on Unix domain sockets to prevent their use by unauthorized users and/or peers are authenticated. TCP/UDP (user datagram protocol) sockets require authentication.
  - **F**: Full containerization of agent code will effectively isolate agents and hide much of the system from them.
  - **R**: An agent may consume too much memory, intentionally or through a programming error, causing the system to become unresponsive and forcing other applications and/or agents to terminate.
  - **M**: A resource monitor is used to place the agent in a memory Linux control group (cgroup) to limit memory usage to what was negotiated before execution.
  - **F**: An agent's out-of-memory (OOM) killer priority can be set lower than critical applications and higher-priority agents so that an OOM condition will cause lower priority agents to be killed first (restricted additions).
  - **R**: An agent may consume too many CPU cycles, intentionally or through a programming error, causing the system to become unresponsive and forcing other applications and/or agents to terminate.
  - **M**: A resource monitor is used to place the agent in a CPU cgroup to limit its CPU utilization to what was negotiated before execution (restricted additions).

User administration of the platform, user interfaces, etc. (Interface 9)

- VOLTTRON supports an easy to use, command-line user interface for platform administration.
  - As part of future VOLTTRON development, based upon user requests, the VOLTTRON team will also develop a simple web management console that runs on the VOLTTRON platform as well as a comprehensive, web-based management system titled VOLTTRON Central.
  - The vulnerability list discussed below is limited to discussion of the command line interface only.



Example Threat and Mitigation

- The platform is locally controlled via a Unix domain socket (9).
  - **R**: Any user with local access to the system has the potential to send command and control messages to the platform.
  - **M**: Access to the control socket is limited by file system permissions on the socket and the owner and group of processes attempting to connect to the socket are validated against an access control list in the platform configuration. The superuser may also be denied access.
- TLS/SSL socket communications, RSA encryption, and x509 certificate management and verification make use of the locally installed OpenSSL library.
  - **R**: Any vulnerability in OpenSSL, such as HeartBleed, could negatively affect the security of the platform and potentially the system.
  - **M**: The system administrator or owner must keep the system up-to-date with the latest security patches, especially with regard to the OpenSSL libraries.

VOLTTRON Platform Hardening Requirements

- Physical Security
- Low Level Device Security
- Boot Security
- Security Updates
- Securing System Access
- Trimming Attack Surface
- Limiting Incoming & Outgoing Network Traffic
- Monitoring system integrity
- Monitoring System State & Resources
- Monitoring and Replicating System Logs

Platform Hardening must be comprehensive to be successful

- Hardening includes:
  - Physical security. Limit who has access to the device. Locked room, locked cabinet with no physical access is preferred. Enable chassis intrusion detection and reporting if possible.
  - Low-level device security. Password protect the BIOS. Ensure periodic updates to keep the BIOS secure. Disable devices that are not needed via the BIOS.
  - Boot security. Restrict boot devices. Disable auto-booting of external devices. Secure the boot loader. Require a password to boot anything other than default kernel.
  - For critical applications, use of a FIPS certified cryptographic module is highly recommended to secure private key material.

Platform Hardening (cont'd)

- Security Updates are required. Configure the system to install the security updates automatically and reboot (if possible) at a particular time. Use the Actuator Agent to reserve the update time window (e.g. 1:30AM on Saturday morning) to prevent other control agents from running.
- Managing system access. Disable all clear text remote system access. No remote root login. Disconnect idle SSH sessions. No FTP, no TELNET, RSH etc.
- Managing users and usernames. Limit number of user accounts. Use two factor authentication if possible. Scan for weak passwords, utilize Linux PAM to strengthen the login process.
- Control incoming and outgoing network traffic
  - Use built-in host-based firewall
  - Rate limit incoming connections to discourage brute force attacks
  - Disable unwanted services.
- Check file system for unexpected changes using Tripwire or similar tool.
- Scan for exploits in the file system using tools such as rkhunter etc.

System Monitoring is a key requirement for Security

- Monitor system state and resources using a tool such as Xymon or Big Brother as well as VOLTTRON Central. Set alerts to notify the administrators if anomalous use of resources is detected.
- Watch system logs and export logs off the system.
  - Logwatch or journalwatch is great for getting daily summaries of system activity.
  - Sending system logs to a remote syslog collector such as Splunk allows long term analysis and trending of data.
  - When logs are available on a remote server, we can inspect the logs even when the local system is compromised
- Use an active intrusion sensor such as PSAD to look for intrusion attempts.

Example Logwatch output

```
################## Logwatch 7.4.0 (05/29/13) ####################
        Processing Initiated: Mon Jul  6 06:25:02 2015
        Date Range Processed: yesterday
                              ( 2015-Jul-05 )
                              Period is day.
        Detail Level of Output: 5
        Type of Output/Format: mail / text
        Logfiles for Host:
##################################################################
-------------------- Cron Begin ------------------------
Commands Run:
    User root:
          cd / && run-parts --report /etc/cron.hourly: 24 Time(s)
       test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily ): 1 Time(s)
       test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly ): 1 Time(s)
---------------------- Cron End -------------------------
-------------------- Kernel Begin ------------------------

1 Time(s): hv_storvsc vmbus_0_2: cmd 0x85 scsi status 0x2 srb status 0x86
1 Time(s): hv_storvsc vmbus_0_2: stor pkt ffff88028e2daf40 autosense data valid - len 20
1 Time(s): storvsc: Add. Sense: Invalid command operation code
1 Time(s): storvsc: Sense Key : Illegal Request [current]
---------------------- Kernel End -------------------------
-------------------- pam_unix Begin ------------------------
cron:
    Sessions Opened:
       root: 26 Time(s)
---------------------- pam_unix End -------------------------
```

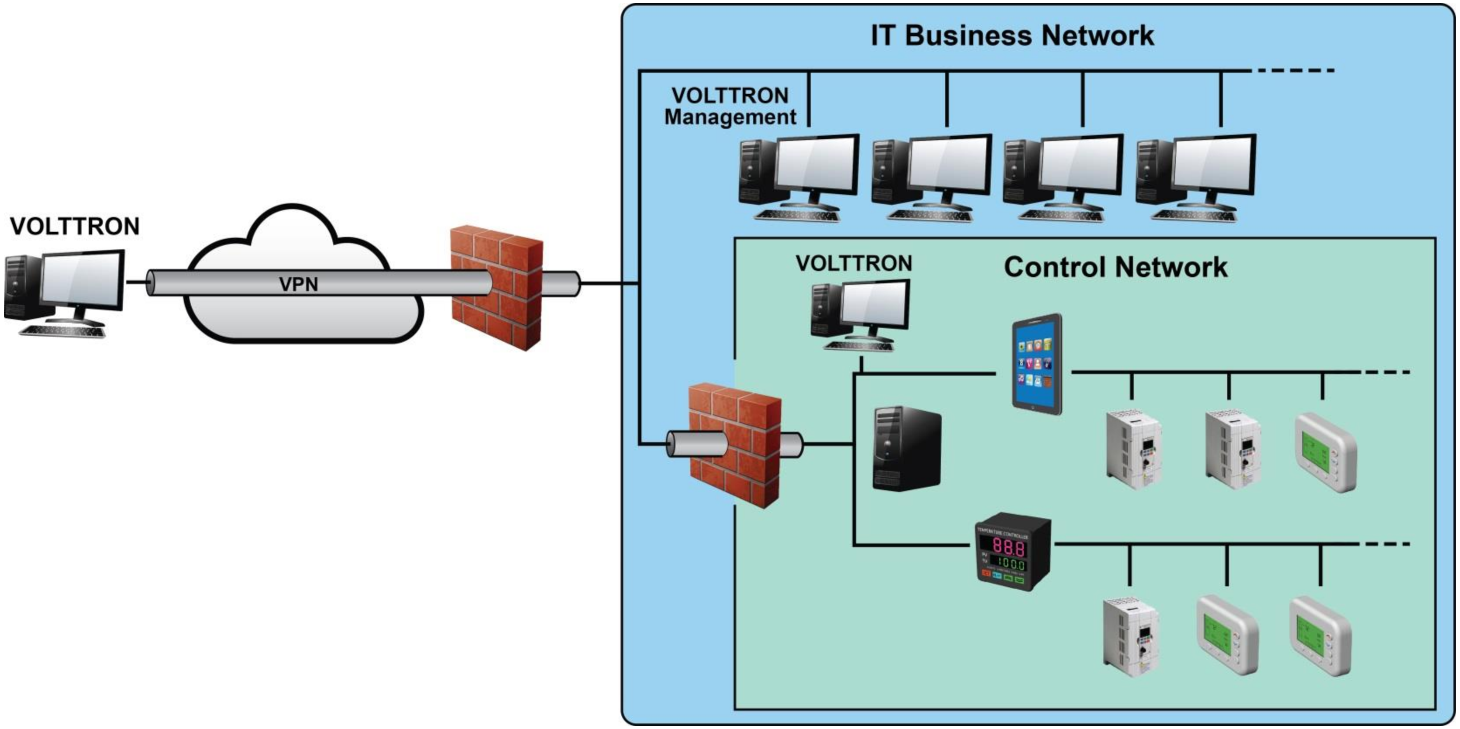
Summary of VOLTTRON Security Features

- Built on Linux to take advantage of its many built-in security features, such as powerful file system permissions, user management, Linux capabilities configuration, control groups, and a highly flexible firewall
- When VOLTTRON accesses remote resources, it is done as securely as possible, utilizing the highest version of TLS/SSL protocols and with the largest key size available to both endpoints. Within VOLTTRON, OpenSSL is used for TLS/SSL encrypted links. The system's OpenSSL libraries are kept as up-to-date as possible to prevent vulnerabilities such as HeartBleed.
- For multi-platform communication, VOLTTRON uses remote ØMQ sockets using CurveMQ elliptical curve encryption.
  - Key exchange: Curve25519
  - Encryption: XSalsa20 stream cipher
  - Authentication: Poly1305 MAC

Summary of VOLTTRON Security Features

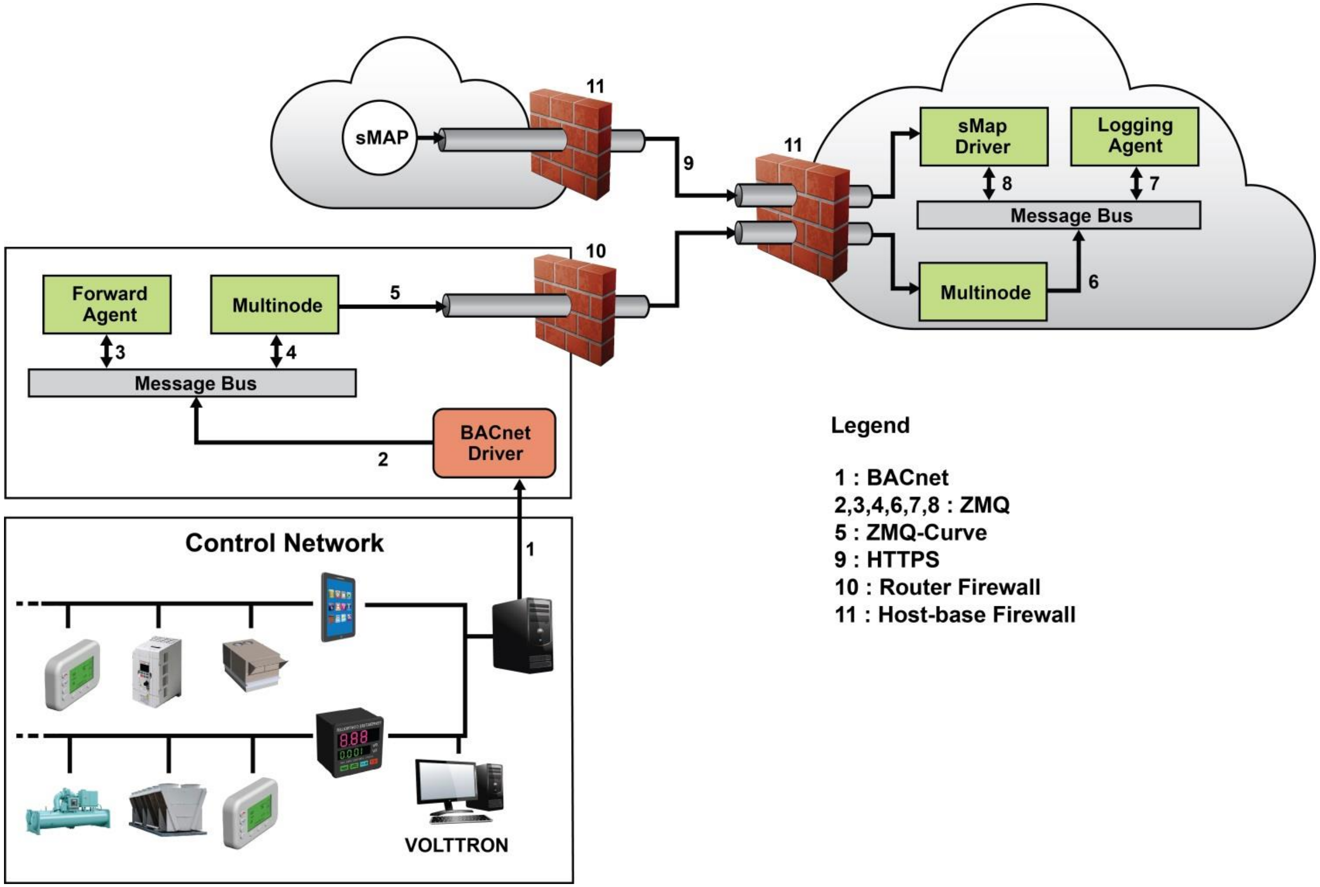
- VIP is used for all internal, inter-agent, and inter-platform communications providing encryption, when appropriate, authentication, authorization, and attribution.
- Linux control groups (cgroups) CPU and memory subsystems are used to limit excessive processor and memory usage.
- Platform control (Unix domain) socket utilizes a mixture of file permissions and access control lists to limit access to authorized users.
- Code is peer reviewed for correctness and security.
- Agent code and packages are signed and verified using RSA encryption with x509 certificates. Unsigned code is not executed unless explicitly allowed by the administrator.

An Example Best Practice for Securing Building Control Networks



- VOLTTRON cannot secure an inherently insecure protocol/network.
- Deployment can help minimize exposure

Example Deployment



Legend

1 : BACnet
2,3,4,6,7,8 : ZMQ
5 : ZMQ-Curve
9 : HTTPS
10 : Router Firewall
11 : Host-base Firewall

QUESTIONS?

- VOLTTRON Resources
  - Wiki: https://github.com/VOLTTRON/volttron/wiki
  - Email: volttron@pnnl.gov
  - Bi-weekly office hours