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

PNNL-SA-106580

# Security Features of VOLTTRON™ Distributed Sensing and Control Platform

Bora Akyol Jereme Haack
Brandon Carpenter

November 2014

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

Pacific Northwest National Laboratory
Richland, Washington 99352

# 1.0 Introduction

VOLTTRON enables rapid authoring and secure deployment of autonomous software agents for distributed sensing and controls. It is designed to be as secure as possible to meet desired security objectives; however, no software can be 100% secure and useful at the same time. Therefore, VOLTTRON uses a threat model approach for determining threats and vulnerabilities of the software and how to reasonably reduce the attack surface and/or harm endured after a compromise.

The first and second releases of VOLTTRON (1.0 and 2.0) focused on:

- protecting the integrity of agent programming through cryptographic means
- protecting agents from using excessive system resources to prevent platform instability
- 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.

This work was completed with the release of VOLTTRON 2.0 and assumes that agent code will be well vetted for correctness and assessed for malicious intent. Currently, this strategy works because of the limited number of VOLTTRON deployments and existing agents are being developed by a small group of trusted developers.

As VOLTTRON usage increases, the number of agent developers is expected to increase, driving the need for additional security features. Future versions of VOLTTRON will see additions to improve agent trust and integrity, including agent message authentication and encryption and full agent “containerization” or “sandboxing.”

This document contains a short summary of the security features implemented in the VOLTTRON platform as of 2.0 release. The discussion will first focus on the threat modeling associated with the platform to put the security features that are provided in context. After the threat discussion, a list of security features is provided as a summary including features planned for future releases of VOLTTRON. The VOLTTRON development team also recommends a thorough review of NIST SP800-82[1] for users of VOLTTRON.

[1] NIST SP800-82 Guide to Industrial Control Systems (ICS) Security accessed at http://csrc.nist.gov/publications/drafts/800-82r2/sp800_82_r2_draft.pdf on November 17, 2014.

## 1.1 Threat Model

Figure 1 below shows two VOLTTRON systems communicating with each other and with external sources. The system on the left side is expanded to show internal components of VOLTTRON and communication with an external historian and a Cloud-based service. Each interface in the figure is numbered so it can be referred to in the text. The approach taken is to discuss security threats associated with each interface. Note that VOLTTRON is built on top of a modern Linux operating system. It should be recognized that while VOLTTRON addresses threats associated with the platform and its interfaces, equal thought needs to be given to the security of the underlying operating system. The VOLTTRON team relies on comprehensive security hardening of the operating system as well as keeping up-to-date with periodically applied patches to further enhance the security of the instances deployed in the field.

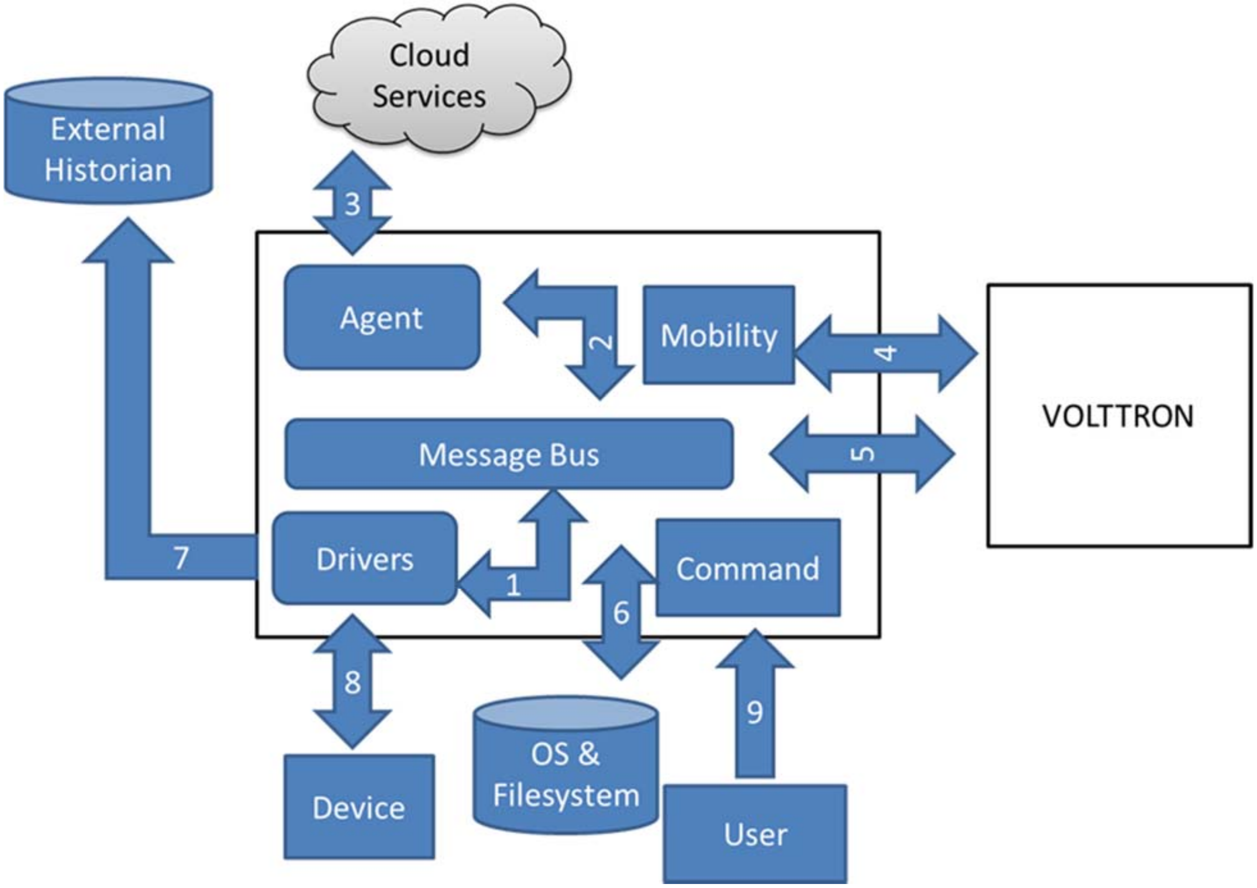

Figure 1: Schematic of VOLTTRON Internal and External Components

The discussion in this section uses the terminology shown below. First, possible attack vectors against the VOLTTRON software are indicated, 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:

1. 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).

### 1.1.1 Vulnerabilities associated with communications between VOLTTRON and other services (e.g. Cloud, etc) (Interfaces 3 and 7)

The VOLTTRON platform allows agents to act as proxies to external resources to move information between a service and the platform (3). In addition, the platform utilizes an external service for storage of data collected from devices managed by the platform and data logged by applications (7). Vulnerabilities associated with these interfaces are listed below:

1. Communication between agents and the Cloud/external historian can 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.

2. Communication between agents and the Cloud/external historian can be stopped by either communications network availability or third party denial of service.

   **R**: A third party can, by means of denial of service attacks or by other means, temporarily interrupt communication between VOLTTRON and external entities. This could result in loss of data or sub-optimal behavior of the control system.

   **M**: VOLTTRON buffers all data that cannot be transmitted to external services. The buffer amount is only limited by the storage available to the platform.

   **M**: VOLTTRON has the ability to go to safe control mode if it cannot communicate with an external controlling entity (that, for example, sets control policies). It is expected that the handling of loss of control policy communication between agents and devices should be handled by the agents.  The VOLTTRON team will provide best practice guidance for agent developers.

3. An external entity communicating with VOLTTRON can be compromised and data coming from or going to VOLTTRON can be intercepted at the non-VOLTTRON end of the communication channel.

   **R**: If an entity that is communicating with VOLTTRON is compromised, then data coming from or going to VOLTTRON can be intercepted or modified.

**M**: All entities that communicate with any control system should be protected as well as the control system itself. If the security of a third party entity is under question, VOLTTRON developers recommend not trusting the entity at all. The VOLTTRON team will provide best practice guidance to protect control systems.

4. Communication between agents and the Cloud (or other outside, network-based) services could be used as an attack vector to compromise the system by intercepting unencrypted or improperly encrypted communications, compromising the remote system, or by performing other attacks.

   **R**: A remote entity could send careful crafted messages to agents which could cause the execution of unauthorized code.

   **M**: Agent code will be reviewed for security issues and malicious intent. Initially (FY15 and FY16), PNNL will review contributions from external contributors before integrating the contributions into the core VOLTTRON code base. Eventually, this review function will be transferred to the organization that maintains VOLTTRON. Agent developers should use encrypted and authenticated services when available and should validate all inputs from other sources before use. The archiver uses TLS/SSL (transport layer security/secure socket layer) when communicating with the archival service.

   **F**: Full agent containerization will limit the effects and reach of a compromised agent.

5. 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**: If an agent is inadequately authenticated or knows the credentials of the remote account, it could send malicious communications.

   **M**: Agent code will be reviewed for security issues and malicious intent.

   **F**: Agents will be sandboxed to disallow unauthorized communications.

   **R**: An agent may exfiltrate data to systems in the Cloud.

   **M**: Agent code will be reviewed for security issues and malicious intent.
   **M**: Firewall rules may be used to limit communications with remote systems.

   **F**: Agents will be sandboxed to disallow unauthorized communications.

### 1.1.2 Vulnerabilities associated with communications between multiple VOLTTRON instances (Interfaces 4 and 5)

VOLTTRON platforms can communicate with each other through the use of the multi-node communication service (5). This service is an extension of the local message bus and forwards messages on a specific local topic to the message bus of another VOLTTRON instance. VOLTTRON also provides a Mobility Service (4), which enables agents to move between platforms and enables administrators to provision VOLTTRON devices in the field. Vulnerabilities associated with these interfaces are listed below:

1. Agents may communicate between the messaging buses of platforms located on different systems (if the multi-node agent is enabled).

   **R**: Agents may subscribe to any topic, without limit, when remote subscriptions are enabled on the remote platform.

   **M**: Agent code will be reviewed for the proper use of remote subscriptions.

   **F**: Topic filtering/authorization will allow limiting remote subscriptions.

   **R**: Agents may publish to any topic, without limit, when remote publishing is enabled on the remote platform.

   **M**: Agent code will be reviewed for the proper use of remote publishing.

   **F**: Topic filtering/authorization will allow limiting remote publishing.

   **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 CurveZMQ protocol to authenticate and encrypt traffic between nodes (when a keyset is provided).

   **F**: May consider replacing the use of CurveZMQ with secure shell (SSH) forwarding/tunneling in a future release.

   **R**: Two additional TCP (transmission control protocol) ports are required for multi-node communication and may be susceptible to denial of service (DoS) attacks. A third port must be opened when the mobility service is enabled.

   **M**: Firewall rules may be applied to help limit the effectiveness of such attacks. VOLTTRON provides no protection itself against DoS attacks.

   **F**: Opening a single port and multiplexing all traffic will limit the number of network ports requiring exposure to the Internet.

2. Agents may move between platforms over the network introducing them to possible man-in-the-middle attacks, spoofing, or other network directed attacks.

   **R**: An unauthorized user might intercept an agent and modify it for malicious use or create their own agent and send it on behalf of a platform they are not authorized to use.

   **M**: SSH tunnels are used to authenticate and encrypt communications between platforms. Agent code is cryptographically signed using x509 certificates. SSH keys are managed using standard SSH configuration files and signing keys are managed using x509 certificates.

   **F**: SSH key usage will be converted to use the x509 certificate infrastructure.

3. JSON-RPC (JavaScript object notation-remote procedure call) function calls are issued over an SSH tunnel to coordinate agents moving between platforms.

   **R**: JSON messages must be completely read into memory. Receiving many extremely large messages could result in a low memory condition. JSON was chosen for serialization because of its simplicity and its unlikeliness to be the source of inadvertent vulnerabilities.

   **F**: A maximum message size could limit memory used. Limiting the number of concurrently parsed messages could help reduce memory usage, but at the cost of slowing communications.

### 1.1.3 Vulnerabilities associated with agent multi-tenancy inside a VOLTTRON platform (Interfaces 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:

1. 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.

2. Agent code runs under the same user account and at the same privilege level as the platform supervisory daemon.

   **R**: A malicious agent can interfere with the platform supervisor, killing it and/or overwriting data. It could also assume the supervisor's role as manager of the communications bus, allowing it to intercept, modify, and/or drop agent communications. This also includes the ability to remove CPU (compute processing unit) and memory limits (restricted additions).

   **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 will effectively isolate and hide agents from the supervisor.

3. Local communication between agents over the message bus is unauthenticated.

**R**: A malicious agent may mimic other agents or the supervisor and send messages on their behalf, potentially causing the creation of unauthentic data or the unauthorized actuation of a device.

**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. Platform will not allow execution of unauthorized agents.
**M**: Operating system and VOLTTRON level features defend against compromise of the internal messaging bus.

**F**: Authentication of inter-agent communications will be supported for validating message authenticity.

**F**: Messaging bus may be enhanced to support policy-based secure access to the messaging bus.

4. 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**: 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. Platform will not allow execution of unauthorized agents.

   **M**: Operating system and VOLTTRON level features defend against compromise of the internal messaging bus.

   **F**: Encrypting the body of inter-agent communications will be supported.

### 1.1.4 Vulnerabilities associated with communicating with and controlling devices (Interface 8)

VOLTTRON drivers (8) allow the platform to both collect data from devices and send control commands. These drivers utilize protocols such as MODBUS, BACnet, or custom-built software to communicate with the device and use the platform's message bus to communicate with agents on the platform. Note that overall security of the underlying communications protocol such as BACnet is outside the scope of this document. VOLTTRON developers recommend use of appropriate cyber security measures to protect the underlying protocol. Refer to the Appendix and the article by Neilson (2013)[1] for a good example on how to secure a control systems network.

1. 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.

[1] Neilson, C. 2013. *Securing a Control Systems Network*. ASHRAE Journal. November 2013.

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.

2. Agents may communicate directly with local devices bypassing the platform drivers.

   **R**: An agent could send commands outside the supervision of the scheduler/actuator causing equipment to operate in an unsafe way.

   **M**: Agent code will be reviewed for security issues and malicious intent before deployment.
   **M**: Agent code is signed (multiple times) and verified before each execution. This prevents an unauthorized third party from tampering with agent code. Platform will not allow execution of unauthorized agents.
   **M**: If underlying protocol supports it, VOLTTRON can implement secure and authenticated communications. For example, for thermostats that support SSL (secure socket layer) communication, VOLTTRON can communicate with the device using SSL.

   **F**: Agents will be sandboxed to disallow unauthorized communications.

3. A device could be physically tampered with.

   **R**: The device could be modified to send inaccurate readings in an attempt to make agents take incorrect and possibly damaging actions.

   **M**: Building and property owners must maintain awareness of their devices to ensure protection and follow best practice guidance described by Neilson (2013) and the guidance provided in the Appendix of this report.
   **M**: Building owners must implement appropriate physical security measures.

   **F**: In an extension of Extending on existing fault detection work, agents could be developed to monitor devices for improper and unexpected behavior.

### 1.1.5 Vulnerabilities with associated with the underlying operating system and file system (Interface 6)

VOLTTRON is built on top of a modern Linux operating system. While VOLTTRON addresses threats associated with the platform and its interfaces, equal thought needs to be given to the security of the underlying operating system. VOLTTRON team relies on comprehensive security hardening of the operating system as well as keeping up-to-date with periodically applied patches to further enhance the security of the instances deployed in the field. The following vulnerabilities (not an exhaustive list) are related to the interface between VOLTTRON and the underlying operating system:

1. Underlying OS (operating system) platform can be vulnerable to attacks because of incorrect or incomplete security update patching.

   **R**: If a platform is not kept up-to-date with respect to security patches, an attacker will be able to compromise the system and gain access.

   **M**: All VOLTTRON systems deployed in the field are configured to automatically download and apply security patches. VOLTTRON developers recognize that in some environments, unattended upgrades are not practical or even can be dangerous. In this case, it is recommended that an engineer apply security updates on a weekly or monthly basis.
   **M**: VOLTTRON systems are protected by appropriate network security measures such as host-based firewalls, intrusion detection, and security monitoring tools to prevent unauthorized access.

2. Underlying OS platform is insufficiently hardened.

   **R**: Security settings of the underlying operating system are not managed correctly and allow overly broad ("loose") access. An example could be the "guest" account, or running unnecessary services.

   **M**: All VOLTTRON platforms deployed in the field by VOLTTRON developers go through a security hardening process that includes turning off unnecessary services, restricting access to necessary services, activating host-based firewall controls and allowing access to the system only by authorized users and hosts.

   **F**: VOLTTRON team will document platform hardening settings as part of the VOLTTRON user's guide.

3. Physical access to the VOLTTRON platform is not controlled.

   **R**: An attacker that has physical access can circumvent many security measures implemented in software.

   **M**: Physical access to any control system must be controlled. There are no exceptions.
   **M**: Linux supports hard disk or volume encryption but this is not sufficient to defend against an attacker that has physical access.

4. 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).

### 1.1.6 Vulnerabilities associated with user administration of the platform, user interfaces, etc. (Interface 9)

VOLTTRON platforms support 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. This document will be updated during the implementation of the VOLTTRON 3.0 features.

1. 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.

2. 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.

3. VOLTTRON is written in Python and runs via the local Python installation.

   **R**: Any vulnerability in Python 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 base Python installation.

4. VOLTTRON uses third-party Python libraries/packages.

   **R**: A vulnerability in VOLTTRON's third-party dependencies could allow VOLTTRON to be compromised.

   **M**: The VOLTTRON developers use mature and actively developed third-party packages with good reputations for stability and security; however, any complex code is likely to suffer bugs that may lead to compromise. While code written in Python is much less susceptible to certain attacks, it is recommended that third-party libraries are regularly updated to the latest compatible version to take advantage of security patches.

## 1.2 Summary of VOLTTRON Security Features

The security features of VOLTTRON are based on the mitigations discussed in the previous section (denoted by M). These security features are summarized below:

- VOLTTRON is 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 first-class firewall.

- When VOLTTRON accesses remote resources 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 CurveZMQ elliptical curve encryption. Keys must be configured for links to be encrypted.

- Paramiko secure shell (SSH) Python package is used to provide RSA encrypted communications between platforms for agent mobility. Most key sizes and encryption scheme defined in version 2 of the SSH protocol (SSHv2) are supported.

- 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.

# Appendix A: An Example Best Practice for Securing Building Control Networks

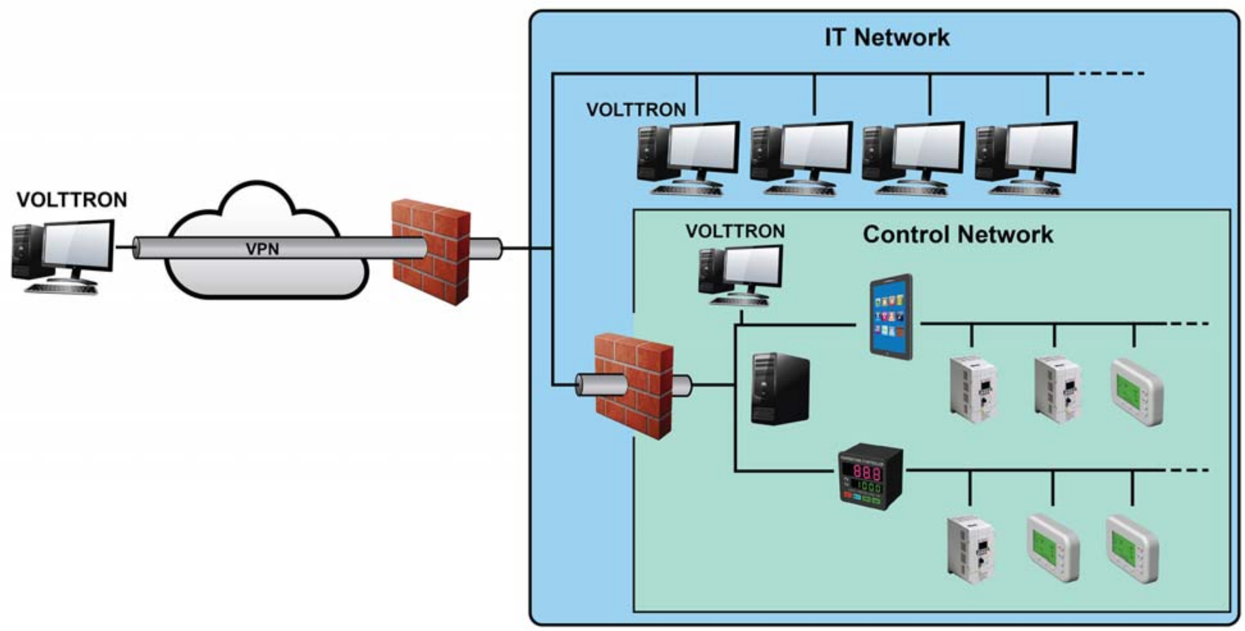

Figure 2: An example best practice of how to secure building control networks. In this example, the control network is completely segregated with its own network and a firewall. The control network is not connected to either the organizational IT (information technology) network or to the Internet directly. This type of segregation reduces the attack points for the building control network. Note that the building control network is protected from the organizational IT network. The access to the control network from the IT network can be tailored by setting up the required firewall rules. Although the graphic above shows two independent networks, this type of security can also be done using virtual local area networks. For more details on this guidance, please refer to Neilson (2013)[2]. Also, note that there is an instance of VOLTTRON running on the segregated building control network and an instance of VOLTTRON running on the IT network. These two instances of VOLTTRON can communicate securely using security features built into the VOLTTRON platform.

[2] Neilson, C. 2013. *Securing a Control Systems Network*. ASHRAE Journal. November 2013.

Pacific Northwest
NATIONAL LABORATORY

U.S. DEPARTMENT OF
ENERGY

*Proudly Operated by* **Battelle** *Since 1965*

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
**1-888-375-PNNL** (7665)

**www.pnnl.gov**