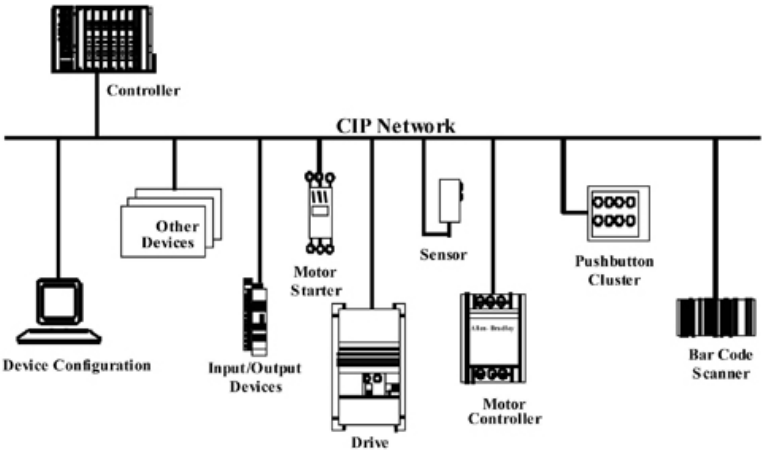
industrial ethernet book
Enhancing Automation and Internet Connectivity

Industrial Ethernet Book Issue 85 / 8

Implementing deep packet inspection for EtherNet/IP

Next generation firewalls with Deep Packet Inspection (DPI) are now mainstream products for IT protocols. Unfortunately, designers and operators of industrial control systems (ICS) have not had access to these advanced technologies to protect their critical communications that involved protocols such as EtherNet/IP.



Various unrelated control system elements all using EtherNet/IP and CIP to communicate.

MISSION CRITICAL CONTROL SYSTEMS need deep packet inspection (DPI) technology even more than IT systems do. By looking at the creation of a DPI firewall for EtherNet/IP and Common Industrial Protocol (CIP), we can explore why DPI is needed for control security, what is available today, and the challenges going forward.

Importantly, there are technical issues in creating an EtherNet/IP DPI firewall that is useable, and emerging solutions that are making these systems more effective in the field.

Potential security crisis

The world's manufacturing, energy, and transportation infrastructures are currently facing a serious security crisis. These critical systems are largely based on legacy SCADA and Industrial Control System (ICS) products and protocols. Many are decades old and were never designed with security in mind.

Yet industry has embraced new network technologies like Ethernet and TCP/IP for ICS. This has enabled companies to operate cost effectively and implement more agile business practices through instant access to data throughout the organization, including the plant floor. While this interlinking improves efficiency, it also significantly increases the exposure of these control systems to external forces, such as worms, viruses, and hackers.

Given the 20 year life cycle that is common for industrial systems, it will be many years before more secure ICS and SCADA devices and protocols are in widespread use. This leaves millions of legacy control systems open to attack from even the most inexperienced hacker. If a hacker or worm can gain access to a control system, it can exploit the protocol to disable or destroy most industrial controllers.

There are a number of possible solutions including encryption and authentication technologies. This article looks at one specific security technology, Deep Packet Inspection (DPI). This advanced filtering technology promises fine-grained control of ICS network traffic, including EtherNet/IP, beyond what is typically found in the IT firewall.

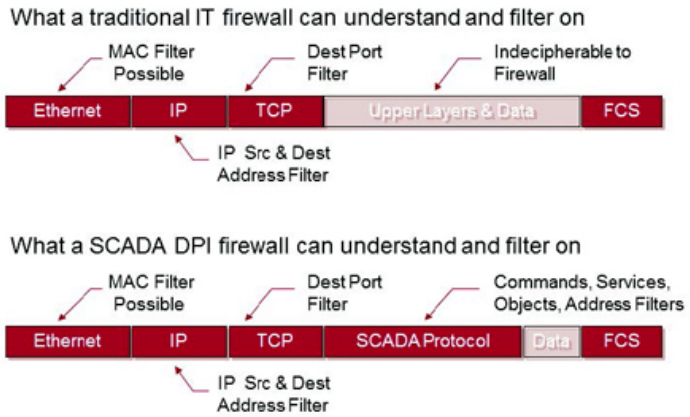
Methods to secure EtherNet/IP

Unfortunately the security issues that are part of the current EtherNet/IP standards and implementations are likely to remain with us for at least the next decade. Discussions are underway to add security features into the specifications, but these are probably several years away.

Even when these changes to the standards are completed, industrial controllers are replaced slowly; their useful lives may be 10, 20 or more years. And the security limitations of the existing SCADA and ICS protocols cannot be addressed through product patches, as their functionality is defined in established standards that take years to change.

Thus it likely will be years before newer, more secure ICS and SCADA protocols and devices are in widespread use. This leaves millions of legacy control devices open to attack from the average hacker. If a hacker or worm can get any control system access, it can exploit the protocol to disable or destroy most industrial controllers.

This means the industry must explore methods to secure these existing protocols and systems, independent of future improvements to the specifications. There are two primary technologies used in the IT world to secure on-the-wire messages. These are encryption and packet filtering. The following sections briefly explore the applicability of both of these technologies to securing EtherNet/IP messages.



Comparing filtering options in a traditional firewall and a DPI Firewall.

Encryption of Data and VPNs

A common method of securing communications is to use cryptographic techniques such as encryption-based tunneling. Usually referred to as Virtual Private Network (VPN) , they are commonly used in the IT world, but are less prevalent in the ICS arena.

VPN technologies create a secure 'tunnel' between two end points over an untrusted network (such as the Internet) by electronically encrypting, authenticating, and validating every message sent between the end points. In other words, VPNs provide three key capabilities:

• Privacy: VPNs encrypt the data passing between the two end points, so that any unauthorized person or device listening to the conversation cannot understand what is being communicated.

• Authentication: VPNs authenticate each end point to the other, so each party in the conversation can be sure that the other party really is who they say they are and is not an imposter pretending to be an authorized party.

• Integrity: VPNs ensure that messages are not modified in transit between the sender and receiver.

While certainly promising in the long term, cryptographic techniques currently suffer from a number of limitations. EtherNet/IP is a time critical protocol and for every packet entering this tunnel there will be overhead incurred to encrypt and decrypt the packet. For devices like PLCs with limited CPU resources, this overhead can result is significant delays. In addition, VPN provides no data validation: if an authorized node on the VPN sends bad data into the tunnel, bad data will appear at the other end.

Probably the most promising use of cryptographic technology is for the authentication of devices, rather than encryption of the data stream. Unfortunately, the issues regarding reliable key or certificate management for embedded control devices like PLCs are still an open research topic.

Firewalls & deep packet inspection

An alternative to cryptography-based solutions is to use firewall technologies to filter network traffic. A firewall is a device that monitors and controls traffic flowing in or between networks. It starts by intercepting the traffic passing through it and comparing each message to a predefined set of rules (called Access Control Lists or ACLs). Any messages that do not match the ACLs are prevented from passing through the firewall.

Traditional IT firewalls apply filters at the TCP and IP layers of a message, using ACLs to check three primary fields in a message:

1. The address of the computer sending the message (i.e., the Source IP Address)

2. The address of the computer receiving the message (i.e., the Destination IP Address)

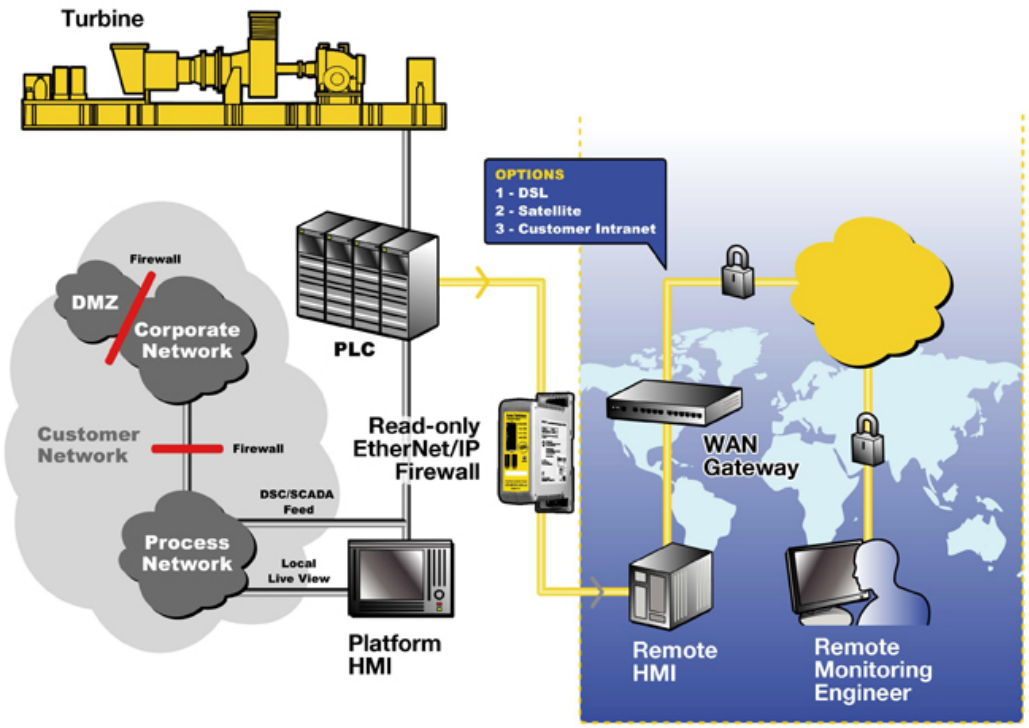
3. The application layer protocol contained by the IP message, as indicated in the destination port number field (i.e., the Destination Port)

The problem with this simple scheme is that it is black and white. One can either allow a certain protocol or block it. Fine-grained control of the protocol is impossible.

This is an issue because the SCADA/ICS protocols themselves have no granularity. From the perspective of the TCP port number, a data read message looks EXACTLY like a firmware update message. If you allow data read messages from a human machine interface (HMI) to a PLC to pass through a traditional firewall, you are also allowing programming messages to pass through. This is a serious limitation.

Deep Packet Inspection (DPI) is an extension to traditional firewall technology that can provide the fi ne grained management of EtherNet/IP traffic needed. Basically, DPI allows the firewall to dig deep into the message to understand exactly what the protocol is being used for. It is designed to understand the specific ICS protocols and then apply filters on fields and values that matter to control systems.

Can DPI technology analyze the message structure of EtherNet/IP packets and then make filtering decisions that would improve the security of a control system? The next chapter of this paper explores this question.



This system example shows the deployment of an EtherNet/IP read-only controller firewall in a natural gas turbine protection system.

Implementing DPI for EtherNet/IP

The architecture of the EtherNet/IP and CIP layers are not trivial and contain many moving parts. Session establishment utilizes a set of commands with varying results and dynamic fields; encapsulation of CIP messaging is done in a tight coupling with the SendRRData and SendUnitData fields on the EtherNet/IP layer.

CIP in itself carries complexity based on the type of connection, whether it is using a message router or Unconnected Send affects the packet structure. Following the connection type a request path embeds the various path segments in 8,16,32 bit forms which then contain logical segments, instance segments, attribute segments, and key segments.

How can you interpret this complexity and design a usable filter mechanism to ensure credible network traffic in an efficient manner? There are two main factors to account for. The first factor is message 'sanity check' actions, which are validation points against the protocol specification. These are used to ensure the structure and values are represented correctly in the packet. The second is the need to identify which fields or actions would make sense (from a user perspective) to be defined as filterable.

Let's start with the sanity checking. There is no shortage of ways in which an attacker could make a PLC unresponsive if not protected correctly. To prevent this sort of Denial of Service (DoS) attack, validation must occur on both EtherNet/IP and CIP layers in tandem. This validation must also take into account directionality of request and response packets as they differ in their format. As noted earlier, there is a bit indicating request or response, so packets can be mapped against source and destination addresses and TCP ports. They can also be checked for correct packet lengths, valid request paths using data segments, valid CIP services and objects combinations, to name a few.

From the user's perspective, sanity checking is executed 'behind the scenes' as it depends on integral knowledge of a protocol implementation. However, sanity checking is still required to be a selectable option because in some cases a vendor may fall short when attempting to adhere to the specification. Ideally, conformance testing should prevent this, but we have observed a number of tested products that vary from the specification in ways only seen on the wire. For example, vendors may interpret a field incorrectly or may implement a special case to meet some design need. Therefore, the ability for the user to turn off sanity checking is important and should be as simple as possible without removing all security.

The next area for filtering is those fields that could be user definable. Since CIP is an object based system with various CIP services acting upon these objects, it makes sense to filter on the fields that denote what object and service is being invoked. Then a knowledgeable user could specify the objects and services that are safe for the firewall to allow, and block all others. For example, to prevent an attacker from modifying the TCP/IP CIP object, the user could remove this object from their "white list" of allowed objects and services. Then the DPI firewall must identify that object in a packet and compare it against the allowed list. If the CIP service or object is not explicitly allowed, then the firewall will block this packet.

The ODVA specification outlines a set of common services and optional object specific services that an object may adhere to. It also allows for vendor specific services. This grouping allows for the abstraction of a read-only filter list or a read-write filter list. As an example, one could group all Set Attribute {Single, All} as write commands, while the Get Attribute {Single, All} could be used as a grouping of read-only commands.

This becomes more complex when looking at the object specific services. If we look against the CIP service 0x54 of a Forward Open when tied to a specific CIP object, this exact same 0x54 on a different object could denote Write Set Value.

This means it is not enough to group CIP services as a whole; you must group CIP services and their partner objects together to develop an abstract grouping of functions. The benefit of creating this object/service pairing is that it provides a user a succinct way of protecting their EtherNet/IP communication stream.

Combine this intelligent filtering of object and service together with sanity checking and you have a powerful tool to not only validate on a per-packet basis but also permit only read functions. Expanding on the idea of combining CIP objects and services together, this filtering list can be dynamic and selectable; there is no limitation in the way the filter list set can be combined. For a specific process it might make sense to allow a read-only function on the TCP/IP object, but permit read-write ability on Analog Output objects. This flexibility allows a user to customize the systems filtering on a per client/server pair basis.

Eric Byres, Erik Schweigert and Michael Thomas *work for Tofino Security, a Belden Brand.*

www.tofinosecurity.com

Source: Industrial Ethernet Book Issue 85 / 8