

Getting security objectives wrong: a cautionary tale of an Industrial Control System*

Simon N. Foley
IMT Atlantique,
Universit  Bretagne Loire, France. simon.foley@imt-atlantique.fr

December 28, 2017

Abstract

We relate a story about an Industrial Control system in order to illustrate that simple security objectives can be deceptive: there are many things that can and do go wrong when deploying the system. Rather than trying to define security explicitly, this paper takes the position that one should consider the security of a system by comparing it against others whose security we consider to be acceptable: Alice is satisfied if her system is no less secure than Bob’s system.

1 Introduction

cite Virgil’s 2014 S+P paper

Contemporary systems are convoluted arrangements of frameworks, software stacks, services and third party components. It is in this complexity, that mistakes are made and that security threats emerge. Despite our best efforts, we continue to have difficulty accurately capturing security objectives, identifying threats and implementing and configuring the security mechanisms that mitigate the threats. The history of the definition of information flow security style properties is a case in point: in the course of 30+ years of research [?, ?, ?, ?, ?], there has been much debate over its meaning and how it might be used in practice. If there can be such variations over what appears to be a conceptually simple security objective—preventing high information from flowing down to low—then what hope have we of

*Draft for the pre-proceedings of the Security Protocols Workshop, Cambridge, 2017.

providing a meaning for security in a convoluted enterprise system, scalability notwithstanding?

Practitioners have tended to take a more operational approach to dealing with security in convoluted systems. Rather than attempting to provide a declarative meaning for security, security objectives are defined operationally. Threats are identified and operational controls are used to mitigate those threats, usually according to some notion of best practice. Thus, the network administrator does not define the meaning of security of an N-tier enterprise network in a declarative sense, rather, the security of the system is defined in terms of its operation: by organising the enterprise network in tiers, the innermost subnet hosts critical data, following best practices.

Security Risk/Threat Management [?, ?, ?] is an example of this operational approach to security, and, while it may scale to convoluted systems, it is in itself convoluted and error-prone. Standards and best practices may help an administrator to identify security risks and deploy defences, however their extensive catalogues encourage a focus on checkbox-style security compliance, rather than security outcomes. At the extreme, approaches such as the Security Content Automation Protocol (SCAP) family of standards [?] champion catalogues with a tremendous amount of detail, leading to challenges in comprehension. For example, the scope for inconsistencies within and between OVAL, CPE, CVE and CCE repositories in SCAP are considered in [?].

Rather than attempting to define security objectives declaratively or operationally, this paper takes the position that one should consider the security of a system by comparing it against others whose security we consider to be acceptable. This is characterised as a refinement relation between systems: Alice is satisfied if her system is no less secure than Bob's system, or, if Alice makes a change to her system configuration then it should be no less secure than her previous configuration. If it can be shown that the refinement relation forms a lattice then its greatest lower, and lowest upper bound, operators provide useful forms of composition. If Alice is happy when her system is no less secure than Bob's system and Clare's system then the lattice join operator ensures Alice has the best possible secure replacement.

In this paper we do not attempt to put forward a general refinement relation (other than it should form a lattice), nor suggest what is meant by a system or security objectives. Rather, we suggest that one should use the notion of refinement as a strategy when considering convoluted systems, or protocols, that have multiple security objectives. This strategy of defining security in terms of comparison has been previously considered for mandatory access control policies [?, ?] and formal security properties [?, ?]. It is

revisited in this paper, where we consider how it can be used in a broader and less-formal setting as a potential approach to dealing with convoluted systems.

This challenges of capturing the full meaning of security in a convoluted system is illustrated in the paper using a ethnographic style study of the connection of an Industrial Control System to the Internet. Shodan.io was used to locate what appeared to be a vulnerable ICS connected to the Internet; the apparent operators of the system were contacted, the vulnerability highlighted and remediation suggested. No further contact was made and Shodan.io was used to track subsequent changes between March 2016 and December 2016. At face value, securing this ICS infrastructure connection should be trivial in terms of network security objectives. However, in our use case we see that there are many objectives to be understood and met, some of which can be contradictory, others are out of the operator’s control, and mistakes are made. Focussing on just the firewall aspects of the system, we illustrate how thinking about the security in terms of refinement may provide a means to deal with this convolution.

2 A convoluted system use case

Despite widespread information on how to defend against infrastructure threats, security can be overlooked or misunderstood when Industrial Control Systems are connected to the Internet. For example, the UK Centre for the Protection of National Infrastructure CPNI recommends that the control network should not be accessible from the public network. Siemen’s S7comm protocol runs over Port 102 and is used for communications between programmable logic controllers and SCADA systems and, at the time of writing a Shodan search found a large number of systems with Port 102 open to the Internet.

In this section we explore one such case: a Siemens SIMATIC S7-300 universal controller that is believed to be used in a public utility installation, as depicted in Figure 1.

Based on CPNI best practice [?, ?] the controller and PLCs should be deployed on an internal control network and a VPN tunnel used when accessing the controller over the Internet/public network. The service should not be directly accessible over a public network. In the following discussion we *speculate* on the threats to which this system might be exposed, based on the information provided by Shodan, vulnerability repositories, and other information in the public domain. Our purpose here is to illustrate the con-

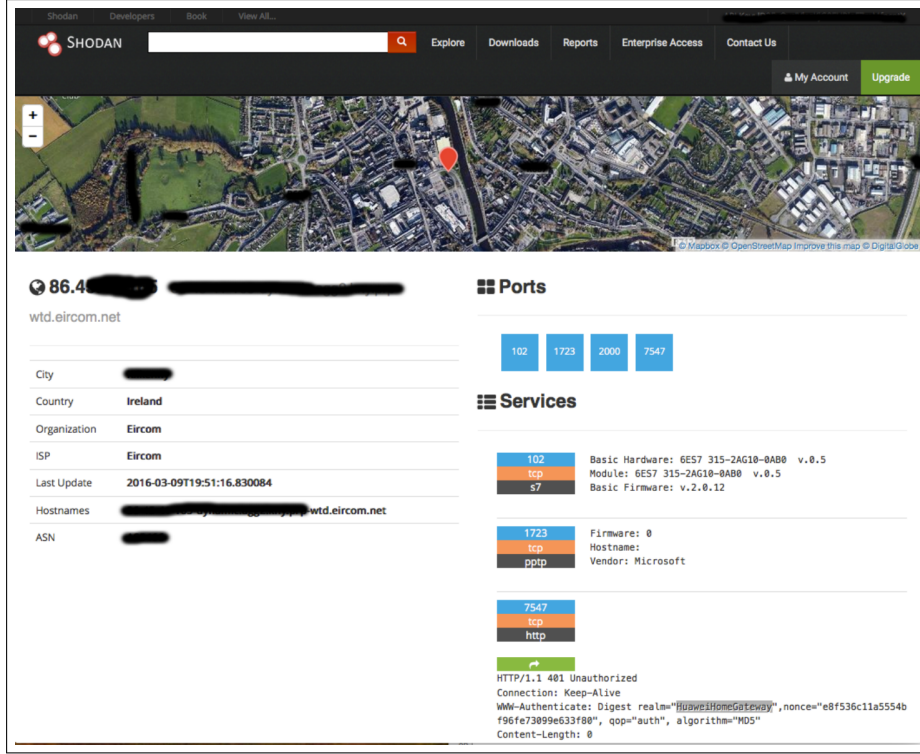


Figure 1: One of the many Internet connected S7 Services.

voluted nature of the system and the security objectives. No attempt was made to access the system nor test our speculation.

CPE An Internet connection is provided via a commercial ISP to what appears to be small outbuildings/storage sheds. Based on the headers (Figure 1), the Customer Premise Equipment (CPE)/gateway router appears to be a Huawei Home Gateway.

It is not evident which particular model is used, however various vulnerabilities have been reported against numerous Huawei Home Gateway models. For example, CVE-2015-7254 (CVSS 5.0) reports *“Directory traversal vulnerability on Huawei HG532e, HG532n, and HG532s devices allows remote attackers to read arbitrary files via a .. (dot dot) in an icon/ URI”*. Huawei routers use the Allegrosoft embedded webserver, which, for example, has had reported buffer overflow vulnerabilities CVE-2014-9223 (CVSS 10.0) and cross-site scripting vulnerabilities CVE-2013-6786 (CVSS 4.3). Huawei [?]

reason that backdoor password vulnerabilities on older Home Gateway models can be mitigated by replacement/identifying them as being at “*End of Service*”.

CWMP The gateway router is deployed with CPE WAN Management Protocol (CWMP) running on TCP/HTTP at Port 7547 of the router. CWMP provides communication between the router and the ISP and supports auto-configuration and management of the router by the ISP.

Running on HTTP, means that the router may be vulnerable to a misfortune cookie attack (CVE-2014-9222,CVE-2014-9223) [?], among others. This vulnerability is a consequence of an HTTP cookie mechanism that allows an attacker to forge session cookies so that its session has administrator privileges. A number of Huawei Home Gateway routers are vulnerable [?] which can be mitigated via a firmware update. However, we note that the installation relies on HTTP digest authentication, which is not generally advised: a remarkable number of routers run CWMP over an unencrypted connection and that an authenticated HTTPS connection would be more appropriate [?], although this is something over which the user has little control.

VPN connection VPN access to the local Control Network appears to be provided via PPTP on Port 1723.

A variety of security vulnerabilities related to using the PPTP protocol have been published over the years [?,?]. Rather than using PPTP, it is suggested, for example, to use OpenVPN or IPSec in certificate mode [?].

Control Network Access to a PLC controller uses the S7comm protocol over TCP/TSAP on Port 102, possibly intended via the VPN service.

While this may be the intended configuration, Port 102 remains open to the Internet, meaning that the controller is directly accessible via the S7Comm protocol from the Internet. This does not follow best practice recommendations [?,?], although the Siemens FAQ [?] could be misinterpreted when it notes that

“[...] if the data is transferred over routers or if firewalls are used, the port must be enabled in the router or firewall according to the service implemented” and recommends that “Port 102 is blocked by default in routers and firewalls and must be enabled for the complete transfer route”. [...]

CVE-2015-2177 notes that versions of the SIMATIC S7-300 is vulnerable to a denial of service attack via this protocol as described by Beresford [?], who also discovered a hardcoded user-id/password (`'Basisk'`) used to access internal diagnostic functions [?]. Based on the header information, we conclude that the SIMATIC S7-300 is a 315-2DP CPU, running firmware V2.6, which has this vulnerability [?]. Our speculation here is that in setting up VPN access, closing direct access to the S7 service via Port 102 was overlooked in the firewall/CPE settings.

Web server Its not evident from the network footprint, nor the documentation, that the SIMATIC S7-300, CPU 315-2DP in the use case incorporates an embedded web-server. Embedded web-servers are supported by some SIMATIC devices, including other versions of the S7-300 and the S7-1200 and examples of other ICS sites serving up SCADA administration from their embedded web server were found. An example of a embedded web server is GoAhead, which lists Siemens as one of its customers [?].

Various vulnerabilities have been published for the GoAhead server, including an application-level (Slow Loris) denial of service attack (CVE-2009-511) and a directory path traversal (CVE-2014-9707). While a software update is recommended to mitigate the directory path traversal vulnerability, it is also suggested that a larger `ulimit` helps defend against the Slow Loris attack. However, this latter recommendation is an example of the need for a trade off, as a larger `ulimit` may make the hosting system vulnerable to a fork-bomb attack.

Changing configuration Based on the geographic location of the system as reported by Shodan (Figure 1), the (likely) Director of IT responsible for the system was informally contacted by email in March 2016. Just the mis-configuration of the S7/VPN and its vulnerabilities were hi-lighted and remediation by blocking Internet access to Port 102 suggested, with reference to the CPNI Best Practices [?, ?]. Receipt of the email was acknowledged, with a response that it would be investigated. No further contact was made and Shodan was used to track subsequent changes between March 2016 and December 2016.

- March 2016. Shortly after the email, the configuration changed with the addition of Microsoft Remote Desktop Protocol on Port 3389, however Port 102 remained open.
- May 2016. The system and all services disappeared from Shodan. This might indicate that the system was successfully configured and VPN

access concealed. However, we speculate that all services, including Port 102, were blocked from the Internet, since it re-appeared the following month.

- June 2016. The system re-appeared, this time with Port 102 (S7), Port 3389 (RDP) and Port 7547 (CWMP) open.
- October 2016. Port 102 was finally closed, however Port 3389 (RDP) was discoverable, displaying a Windows login with a specific user-id and prompting for a password.
- December 2016. The system and all services disappeared from Shodan. The hope was that Port 3389 (RDP) was successfully concealed, however in
- March 2017. The system re-appeared, this time with Port 2000 discoverable upon which a “RemotelyAnywhere” login prompt was available over HTTPS. The corresponding certificate used a SHA1 based signature, necessitating a security exception by the browser.

3 What’s the likely setup behind the scenes?

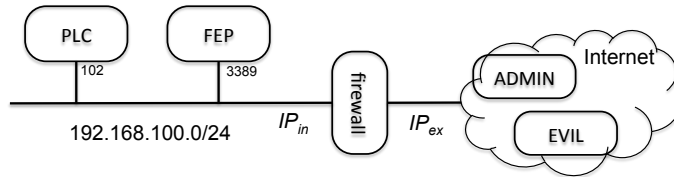


Figure 2: A network connected ICS

The use case is intended to illustrate the convoluted nature of a contemporary system and how easily mistakes are made in achieving security objectives. Informed by this, we give a simplified interpretation of how deploying the VPN went wrong. The configuration is depicted in Figure 2 where a PLC/controller (Port 102) and Front End Processor FEP (Port 3389) are on an internal network, behind a firewall. Access is required by the external enterprise/administrator (ADMIN), but the attacker (EVIL) should not have access.

UPol In the initial configuration, following Siemens FAQ8970169 “Port 102 is blocked by default in routers and firewalls and must be enabled for the complete transfer route” and naively setting up a VPN gives rise to the following firewall policy.

Index	[...]	Src IP	Src Port	Dst IP	Dst Port	Action
1	...	*.*.*.*	≥ 1024	PLC	102	ALLOW
2	...	*.*.*.*	≥ 1024	FEP	3389	ALLOW

For ease of presentation we assume that RDP provides the VPN from the outset.

CPNI CPNI recommends “SCADA communications should be encrypted and routed through a VPN tunnel through corporate IT or other non-critical networks.” is implemented as the policy:

Index	[...]	Src IP	Src Port	Dst IP	Dst Port	Action
1	...	*.*.*.*	≥ 1024	PLC	102	ALLOW
2	...	*.*.*.*	≥ 1024	FEP	3389	ALLOW

RPol Access to the VPN should be limited to authorised IPs:

Index	[...]	Src IP	Src Port	Dst IP	Dst Port	Action
1	...	ADMIN	≥ 1024	FEP	3389	ALLOW
2	...	*.*.*.*	*	FEP	3389	DROP

Composition I A common strategy for managing firewall policies is to compose policies in sequence. On understanding that policy *CPNI* must be enforced, the policy is revised as *UPol*; *CPNI* and then further extended to *UPol*; *CPNI*; *RPol* in order to limit RDP access (Table 1). The resulting anomalies whereby *CPNI* and *RPol* are shadowed/redundant by *UPol* mean that we do not achieve the *CPNI* nor *RPol* objectives.

Composition II Having realised the mistake, the administrator revises the policy to enforce the *CPNI* policy objectives first, followed by the remaining policy, that is, *CPNI*; *RPol*; *UPol* (Table 2). However, while blocking Port 102, there is an anomaly between Rules 2 and 4, which means that the RDP objective that only ADMIN should have access to the FEP is not enforced.

Index	[...]	Src IP	Src Port	Dst IP	Dst Port	Action
1	...	*.*.*.*	≥ 1024	PLC	102	ALLOW
2	...	*.*.*.*	≥ 1024	FEP	3389	ALLOW
3	...	192.168.100.0/24	≥ 1024	PLC	102	ALLOW
4	...	*.*.*.*	*	PLC	102	DROP
5	...	external	≥ 1024	FEP	3389	ALLOW
6	...	ADMIN	≥ 1024	FEP	3389	ALLOW
7	...	*.*.*.*	*	FEP	3389	DROP

Table 1: Composition *UPol* ; *CPNI* ; *RPol*

Index	[...]	Src IP	Src Port	Dst IP	Dst Port	Action
1	...	192.168.100.0/24	≥ 1024	PLC	102	ALLOW
2	...	*.*.*.*	*	PLC	102	DROP
3	...	external	≥ 1024	FEP	3389	ALLOW
4	...	ADMIN	≥ 1024	FEP	3389	ALLOW
5	...	*.*.*.*	*	FEP	3389	DROP
6	...	*.*.*.*	≥ 1024	PLC	102	ALLOW
7	...	*.*.*.*	≥ 1024	FEP	3389	ALLOW

Table 2: *CPNI* ; *RPol* ; *UPol*

Composition III The administrator tries another re-arrangement of the policy as *RPol* ; *CPNI* ; *UPol* (Table 3) which happens to be anomaly-free and meets our objectives. However, re-arranging policies in this manner so that they are anomaly-free does not necessarily achieve our objectives, especially when policies may run to many thousands of rules.

4 Comparing configurations

4.1 Refining firewall policies

In the following we use a refinement relation for firewall policies that is a simplification of the more general firewall algebra described in [?,?]. In particular, packets are modelled in terms of source and destination IP addresses and Ports:

$$Packet == IP \times PORT \times IP \times PORT$$

Index	[...]	Src IP	Src Port	Dst IP	Dst Port	Action
1	...	ADMIN	≥ 1024	FEP	3389	ALLOW
2	...	*.*.*.*	*	FEP	3389	DROP
3	...	192.168.100.0/24	≥ 1024	PLC	102	ALLOW
4	...	*.*.*.*	*	PLC	102	DROP
5	...	external	≥ 1024	FEP	3389	ALLOW
6	...	*.*.*.*	≥ 1024	PLC	102	ALLOW
7	...	*.*.*.*	≥ 1024	FEP	3389	ALLOW

Table 3: Composition $RPol$; $CPNI$; $UPol$

where (ip_s, p_s, ip_d, p_d) is a packet from source IP ip_s , port p_s , with destination IP ip_d and port p_d . A firewall policy $P : Policy$ defines a set of packets $accepts(P)$ that are accepted and a set of packets $denies(P)$ that are denied/dropped. We have

$$Policy == \{Accepts, Denies : \mathbb{P} Packet \mid Accepts \cap Denies = \emptyset\}$$

and we assume a default deny for packets not referenced by P . This is a very simple representation of an anomaly-free policy that is adequate for our purposes; our discussion here can be extended to the more general firewall algebra [?, ?] that supports policies with IP and port ranges, and numerous other attributes. Two types of constructor provide a simple policy notation.

Weak allow/deny. Packets not in $X : \mathbb{P} Packet$ are default deny:

$$Allow\ X = (X, \emptyset); \quad Deny\ X = (\emptyset, X)$$

For example, permit S7 traffic from the internal network (IP_{in}) to the PLC:

$$Allow(IP_{in} \times PORT \times \{PLC\} \times \{102\})$$

Strong allow/deny. Packets not mentioned in X are explicitly denied/accepted:

$$Allow^+ X = (X, Packet \setminus X); \quad Deny^+ X = (Packet \setminus X, X)$$

For example, block external S7 traffic to PLC, everything else permitted:

$$Deny^+(IP_{ex} \times PORT \times \{PLC\} \times \{102\})$$

Policy Replacement. Policy Q can be replaced by policy P , if $P \sqsubseteq Q$, that is, P is no less restrictive than Q . For all $P, Q : Policy$:

$$P \sqsubseteq Q \Leftrightarrow (accepts(P) \subseteq accepts(Q)) \wedge (denies(P) \supseteq denies(Q))$$

The most restrictive policy is $\perp == (\emptyset, Packet)$ and the least restrictive policy is $\top == (Packet, \emptyset)$ and we have $\perp \sqsubseteq P \sqsubseteq \top$ for any policy P .

Policy intersection and union. The least restrictive safe replacement for P and Q is $P \sqcap Q$, where

$$P \sqcap Q == (accepts(P) \cap accepts(Q), denies(P) \cup denies(Q))$$

The most restrictive policy that can be safely replaced by P or Q is $P \sqcup Q$:

$$P \sqcup Q == (accepts(P) \cup accepts(Q), denies(P) \cap denies(Q))$$

The set *Policy* is a lattice under partial order \sqsubseteq , greatest lower bound operator \sqcap , and lowest upper bound operator \sqcup [?, ?]. Figure 3 gives an example of some policy orderings.

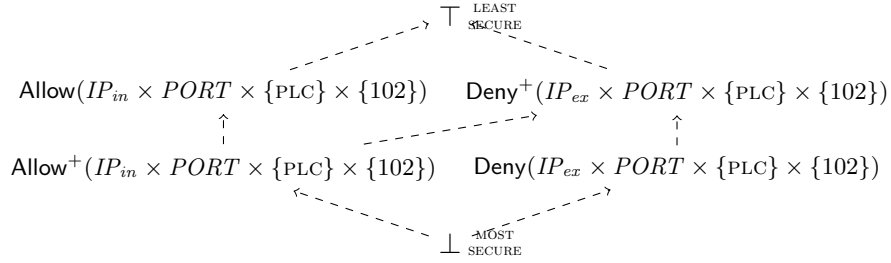


Figure 3: Some policy orderings

4.2 Comparing the ICS firewalls

CPNI recommendation. The CPNI objective specifies that internal S7 traffic to the PLC is permitted while external traffic should be blocked but no constraints on other external traffic.

$$CPNI == Allow(IP_{in} \times PORT \times \{PLC\} \times \{102\}) \sqcap Deny+(IP_{ex} \times PORT \times \{PLC\} \times \{102\})$$

Remote Desktop recommendation. The RDP objective permits administrator VPN access to the Front end processor, and to deny all other traffic.

$$RDP == \text{Allow}(\{\text{ADMIN}\} \times \text{PORT} \times \{\text{FEP}\} \times \{3389\}) \\ \sqcap \text{Deny}^+(IP_{ex} \times \text{PORT} \times \{\text{FEP}\} \times \{3389\})$$

For any packet the *CPNI* or *RDP* policy should apply. Therefore, if the initial policy was *Pol*, then changing it to incorporate CPNI on a RDP based VPN gives:

$$Pol' == Pol \sqcap (CPNI \sqcup RDP)$$

and since $(Policy, \sqsubseteq)$ forms a lattice the administrator can be sure that the new policy *Pol'* a safe replacement of the original policy, and the new security objectives. Furthermore, it is the best secure replacement under \sqsubseteq .

5 Conclusion

In the spirit of Jackson [?] we have related a story about an Industrial Control system in order to illustrate that simple security objectives can be deceptive: there are many things that can and do go wrong when deploying the system. We suggest that rather than trying to reason about the security objectives explicitly, we capture them indirectly by comparison in the form of a refinement relation. In developing this idea, we limit ourselves in this paper to just firewall policies which is a homogenous collection of objectives. However, the strategy can be extended to heterogenous objectives and we are currently exploring how other, non-firewall, aspects of the ICS use case can be incorporated into the refinement relation.

Acknowledgement This research was supported in part by the CHIST-ERA project *DYPOSIT: Dynamic Policies for Shared Cyber-Physical Infrastructures under Attack* and by the Cyber CNI Chair of Institute Mines-Télécom which is held by IMT Atlantique and supported by Airbus Defence and Space, Amossys, EDF, Orange, La Poste, Nokia, Société Générale and the Regional Council of Brittany. It has been acknowledged by the French Centre of Excellence in Cybersecurity.

References