Getting security objectives wrong: a cautionary tale of an Industrial Control System (Transcript of Discussion)

Simon N. Foley

IMT Atlantique, LabSTICC,
Université Bretagne Loire, France.

Simon Foley: This is work that evolved by accident. Last year I started a project on industrial control system security, and by way of educating myself about the kinds of things that can go wrong, I used Shodan to search for an existing Industrial Control System connected to the Internet. I gave my first version of this talk in March 2016, and have given it a couple of times in the interim. Each time I prepared for the talk, I revisited the ICS, and each time its configuration had changed. This talk is what I learned from that experience.

Firstly, a short overview of the motivation for this talk. My focus is on Industrial Control Systems that are connected to public networks, with one very simple security objective, provided by the UK Centre for the Protection of National Infrastructure, which is that any SCADA communication with the ICS over a public network should be encrypted and routed through a VPN tunnel. At face value this seems like a straightforward security objective. To investigate, I used Shodan to search for ICS systems connected to the Internet with Port 102 open. Siemens use this port to host their S7 protocol for SCADA control packets, enabling a system to interact with the control system over a network. The search found many such systems, connected to the Internet and willing to speak S7 on Port 102. I chose one such connected ICS system. For my tale, its not important where it was: it looked like it was owned by a public organisation.

Studying the Shodan report for this system, we discover that the S7 service may be vulnerable to a denial of service attack and may have a default backdoor userid/password. In being connected to the Internet with Port 102 open, in principle anyone could exploit these vulnerabilities and compromise this system. Of course, I didn't try to access the system; I viewed it through the eye of Shodan. The deployment of the system appeared not to meet the CPNI security objective, which should be to send all this SCADA traffic encrypted over a VPN and block the service from direct access over the public network[1]

Perhaps the engineer deploying the system followed the CPNI objective and installed the VPN, however he or she didn't realise that securing the system was much more complex and would require multiple objectives to be met. The question then is this: what exactly are the security objectives for a simple system

[1] Shodan reported other services available, including PPTP and CWMP, and the reader of this transcript is directed to Section 2 of the associated paper for further details. In short, it illustrates that security is not as simple as ensuring that just one objective is met.

like this? From even our cursory look at the system using Shodan, it is apparent that there are multiple security objectives. The further we drill into the details of the system the more security objectives we are likely to have to consider. Its not likely that the typical engineer deploying an ICS will have this understanding.

How might a security expert approach the problem of identifying the necessary security objectives in this case? Many of us have looked at how to define and reason about security objectives in terms of formal security properties; our view of the system will be necessarily simple and we hide much of the complexity to make the modelling tractable. Whatever about agreeing on how these properties should be specified, we will very likely overlook security objectives for the actual system deployed. For example, if our ICS was a potato processing plant then the reality is closer to this Heath Robinson contraption[2] for peeling potatoes than to a Jonas potato peeler. Faced with this particular PLC, the commercial router box and all the legacy services and software that go with it, then how do we define what is meant by security in this case? We know that this is difficult and it is more likely that the efficacy of the deployment will be based on the expertise of the individual setting it up, best practises, and so forth. Thus, rather than attempting to directly define what we mean by security in terms of exact objectives, we should instead, consider it indirectly in terms of comparison

If Alice believes that Bob's system is secure then she would like configure her system to be like his. Alice does not need to know what it is about Bob's security objectives that make his system secure, other than to know that her resulting system is no less secure than Bobs. We capture this in terms of a security comparison relation, whereby $P \sqsubseteq Q$ means that security objective P is no less secure that security objective Q. This gives us a basis for secure replacement. If Alice currently requires security objective Q then she can safely replace it by a P, where $P \sqsubseteq Q$, and then have P as her new objective; perhaps the result is more secure, but regardless it is no worse in security terms than her original requirement. The idea then is that instead of defining security explicitly, for example, in terms of the exact objectives of Alice, we define it implicitly, by comparison, that Alice's objectives should be no less secure than Bobs.

Joan Feigenbaum: I don't understand how you can prove that one is more secure than the others who don't define their security.

Reply: Its a question of semantics. We still have define security, implicitly, using the comparison relation. However, we do not then have to explicitly define the security objectives for Alice, other than they are no worse than those of Bob. Alice's new objectives P will block more packets than her previous objectives Q.

Vashek Matyas: Maybe the question is not about security, but compliance?

Reply: I agree, compliance fits within this view. Alice would like to replace her objectives by something stronger which ensures the CPNI requirement that traffic is encrypted over a VPN; if $CPNI \sqsubseteq P$ holds then Alice is happy since her new objective $CPNI$ is no less secure than her previous objective P. She's not checking that she's just compliant with some CPNI objective, but also that her new objective does not break whatever other objectives she might have.

[2] A cartoon entitled *The Professor's invention for peeling potatoes* by Heath Robinson.

It is reasonable to expect that secure replacement should form a partial order over the set of objectives. If we can show that the ordering relationship also forms a lattice, then we get intersection and union operations which gives us a way not only to compare our objectives, but to also compose them. So for example, if I have two objectives P and Q, and I want my system to uphold these two objectives or at least I want an objective that's suitable replacement for P and Q, then I'll take their intersection. Intersection, as a greatest lower bound operator, gives us the most flexible/least restrictive objective that is a secure replacement for P and Q. This is a much more useful objective than, for example, the most restrictive objective that permits nothing and is also a secure replacement for P and Q; however, at the bottom of the lattice it is overly restrictive.

Bryan Ford: The example you used to motivate this I think really nicely illustrates my question. One of the problems with this approach, it seems to me, is that if you have an ideal model, for example that requires that the only traffic emerging from this SCADA system has to come out of the VPN. But the only possible way to implement the VPN is that at a lower of level of abstraction, at the implementation level, packets do have to emerge out of the physical device and that point where they emerged out of the physical device was exactly one of the places you point out where the implementation failed to be more secure.

Reply: I have some examples later which hopefully answer that.

Bryan Ford: Okay, I'll let you get to that then.

Paulo Esteves Verissimo: If we're considering that we can substitute secure parts of the system with other more secure parts then how do you deal with emergence? In this case it may turn out that while a replacement part may be more secure than the part it replaces, some new behaviour may emerge, that breaks security, as a result of the interoperation between this new part and the rest of the system.

Reply: That's an important point. Security is not a functional property and we know that the composition of two secure systems is not necessarily secure. For my purposes, I assume that this is dealt with by the context in which you define and use the comparison ordering relation. Thus far I have only considered objectives; it becomes a much harder problem if we are to also consider the security controls that are intended to meet those objectives. If security controls A meet objective P and security controls B meet objective Q then we would need to be very careful defining how to compose the systems/controls A and B if we wanted their composition to meet the intersection of the objectives. I show a simple example of this working for a firewall control later, but for arbitrary collections of controls it will be a challenging problem.

I don't want to prescribe a specific secure replacement operation. If you're dealing with multiple objectives then perhaps what you should do is think about them in terms of "what does it mean to compare the objectives?" and see is there a partial order in it, and whether it form a lattice. Because if it does form a lattice, then it gives me an algebra that I can use to compare and compose my objectives. And this is hard. We've done it for a secure replacement ordering for iptables and it was a lot of work just for something as simple as a firewall policy.

Virgil Gligor: When you talk about security objectives you implicitly have in mind some sort of a definition of an adversary. Unless you have some sort of adversary, the security objectives are useless, right?

Reply: Yes.

Virgil Gligor: Now if you have a correct and complete definition of an adversary, it turns out that you might get your lattices. And, in fact, I have a bunch of examples in a paper presented at 2014 IEEE Security and Privacy Symposium about how to define an adversary completely and correctly such that you can measure such you can have a lattice. So having said that, I understand that your problem here is much harder than that. Because you have to combine not only the adversary definition in some way to your security replacement but you also functionally have to combine it with protocols that are really fine in some sense. So it's integration that causes the composition problem.

Reply: That's a good point because what happens then is that, for all the attraction to be able to say, "It's a lattice," and you'll end up with this problem again.

Virgil Gligor: Exactly.

Reply: The question is then one of "What's the right level of abstraction?". I don't want to build a formal model of the entire system's behavior with a Dolev-Yao style attacker built into it because, good and all as it is, unless we abstract away detail we'll end up with the professor's potato peeler, the Heath Robinson contraption that's so complex that we've likely overlooked something. And the same applies to compliance, even though the aspiration of compliance is good. Again, the problem is one of complexity. If you have an organisation that has tens or even hundreds of thousands of security controls for which you've got to check compliance, then you're back to this problem of complexity again and trying to figure out what your security objectives are. So we've got to choose some level of abstraction to work with. And further, instead of trying to explicitly define what is meant by security I suggest we should implicitly define it using comparison.

With this in mind, we return to the cautionary tale about the Industrial Control System, an apparently simple system connected over the Internet. Based on the Shodan reports I conjecture what happened behind the scenes; it is speculation, but regardless, it does make for a nice way to illustrate the challenge of dealing with multiple security objectives.

We likely have a local network with a PLC/SCADA system with Port 102 open for the control system traffic and a front end processor FEP. These are behind a firewall/router with a VPN tunnelling service on Port 3389, using the remote desktop protocol. On the external network/Internet we have an Administrator and the attacker Evil. We imagine that the engineer deploying the system, set up the VPN to tunnel traffic to Port 102. That's all good, however, its complicated because the engineer may also have consulted the Manufacturer FAQ[3] where they advise that "Port 102 is blocked by default in routers and firewalls

[3] On being alerted to this issue, the advice was removed from the Manufacturer's website.

and must be enabled for the complete transfer route". If the engineer follows this advice then they may reconfigure the firewall to permit traffic to Port 102, despite the presence of the VPN and the CPNI recommendation. We have an apparent conflict between objectives.

When I first looked at the Shodan report for this system, I emailed the people who I believed operated this system. I pointed out to them that they hadn't set up their VPN correctly, gave them the documentation on the vulnerabilities, noted that they should to setup their network/VPN correctly and pointed them to the CPNI information on best practices. They responded that they would investigate. That was in March of last year (2016) and later that month, when I consulted Shodan they had changed the configuration of their system. Their change was to add a further VPN service running on a new port, but they didn't disable the previous VPN service nor block off Port 102. A new security objective was added, but they didn't address the previous objectives/setup which were flawed. Over the intervening months Shodan reported various changes on how the system was connected to the Internet and the services that were exposed[4].

Fabio Massacci: Could not this be the case that they were installing other things, then at the central point of the click through process it just kept accepting these things though the firewall?

Simon Foley: Yes, it could be, although SCADA systems have a reputation for having poor update management. Nevertheless and however we interpret things, it does illustrate how easy it is for us to get the objectives wrong. Even though the rules appear quite straightforward, if were not paying attention, how do we compose them correctly?

Fabio Massacci: Yes but my challenge is whether they actually know what the objectives are.

Reply: Indeed, what are the objectives!

Fabio Massacci: The system didn't work previously for them, they click through a notice from windows that says we want launch through the firewall, they tried it didn't work, they redid it again, now it works, now they have the login and password screen. This is actually a feature; it not a bug, it's taking you directly to login.

Reply: Yes. What they likely want in this case is "I'd like to replace my current configuration with the system is no less secure than the previous configuration," regardless of how the installation might work.

The cautionary tale is about misunderstanding of firewall objectives. In another piece of work we have developed an algebra for constructing firewall policies. This algebra, provides a lattice ordering which gives us a definition for secure replacement. A firewall Policy Q can be replaced by another firewall policy P, if P is no less restrictive than Q, that is if all the packets accepted by P are accepted by Q and all the packets denied by Q are denied by P. This is a simple definition; the original paper provides a full comparison operation for

[4] The reader of this transcript can find the account of these changes and their security implications in Section 4 of the accompanying paper; they provide the tale on how difficult it can be to get multiple security objectives right.

IPtables firewall policies and we managed to prove that it forms a lattice with a greatest lower bound and lowest upper bound operators. This means that we can compose firewall policies while providing secure replacements.

Ross Anderson: That presumably assumes that the firewall policies are all stateless.

Reply: We can manage some information about stateless policies also.

Ross Anderson: I had a research student who looked at what access control meant in Software Defined Networks. And it turns out that access control and VPN is pretty much the same as Cheswick-Bellovin firewall rule. So you could port across what we know about access control from the past 60 years and compare it to the last 25 year's worth of firewall rules. The interesting thing is that what happens to state, which we deal with differently.

Simon Foley: We had thought about how the idea of a lattice of firewall rules might be adapted to Software Defined Network routing rules. Building a similar algebra for SDN would be tricker than the algebra for single firewalls, but it should be possible. In another paper we had looked at modelling multiple firewalls in terms of compositions of upstream/downstream policies.

That's more or less what I wanted to talk about today. We have convoluted systems where there's many pieces to the many components, many people involved, many objectives, lots to go wrong. And as a way of trying to deal with this complexity is, rather than trying to define directly what we mean by security, defining it by comparison instead. Alice is happy if her system is no less secure than Bob. There's a simple example of the firewall algebra in the paper, but we reference another paper, which gives all of the details.

Lastly, I have a question for the audience about the ethical considerations for this kind of work. As I mentioned at the start, it was quite accidental that I ended up studying this particular SCADA system over a period of a year. It was only after that I'd written the paper that I started to think about the ethical considerations. The British Psychological Society gives some recommendations on ethics and conduct. Which is, "unless informed consent has been obtained, one should restrict research based on observations of public behaviour to those situations in which persons being studied would reasonably expect to be observed by strangers, with reference to cultural values" and so forth. The users in this case connected their system to the public network, albeit incorrectly. Presumably, they had an expectation that they were doing all of this in private and that they wouldn't be observed. Is it ethical for me to have carried out this ethnographic style study of the behaviour (albeit through the eye of Shodan) of these users without their knowledge, and do research and write some commentary on it?

Frank Stajano: Yes. I would say that the fact that you are doing research puts you in a safe place, both legally and ethically, because they have connected their system to the network, any bad guy can see that, why should they think that the good guy can't see what the bad guys can see.

Ross Anderson: I think I would take a different view of it. I'm on our University Research and Ethics Committee and this comes up again and again. The reason that Cambridge has a school-level ethics committee as well as a unified one, for example as CMU does, is that if you try and apply the psychologist rules and the clinical medics rules to ours, we can't get any work done at all.

There's a second issue that there is a legal expectation of privacy as the Mirror vs. Campbell case showed, that even celebrities behaving on the public street can sometimes have an expectation of privacy. So I would say that users in your study could say that, yes, you violated something by Shodaning them, but whether that was the privacy of any identifiable individual is another question.

Reply: Yes, apart from the user-identifier revealed in the screenshot of the RDP login prompt. The target of this study is a public organisation, so there is a duty of care on them, that if they're going to connect to the network, then they do it properly. Does the public expectation that they should be competent competency outweigh privacy?

Fabio Massacci: Was there any accident in the meanwhile?

Reply: No that I know of.

Fabio Massacci: So this means they were competent enough? Forget about the part about the vulnerability and so on, from our perspective, they were competent enough.

Frank Stajano: There was no *known* incident.

Fabio Massacci: Sure, but that's the point, right? If there was some incident of serious consequences, it would have ended up in the news, we'd find something on the newspaper. But everything was fine.

Paulo Esteves Verissimo: I would opt for something in the middle. You don't want to protect yourself legally completely, you don't want to be very stringent. I think I have the right to publish it. And if you have something to say about it and if you want to fix it.

Bryan Ford: I'd like to go back to the actual content of your topic of your, as opposed to the ethical issues surrounding, which are also interesting, I agree. Your example is interesting and a nice example of what goes wrong, for example, if you try to reason about security policies in the form of linear allow, deny lists. But it's not that surprising that linear allow/deny lists are a bad way of reasoning about security policies. On the other hand, it seems, completely implausible that the average person setting this kind of thing up is going to be able to reason in formal reasoning system. At the same time, do you know anything in the middle that's better than allow/deny lists which are incomprehensible to the average Joe, at least?

Reply: We have our IPtables algebra, and while we do define policies in terms of allows and denies, we don't have to worry about how we sequence them. We use the join and intersection operators to build policies that are suitable secure replacements for previous policies. We could build a tool where the user thinks not in terms of rules, but in terms of compositions of policy objectives. In the case of our study, the engineer would take the original policy and extend it by composition to include the CPNI objective; the result will be a secure replacement for the previous policy so there'll be no security surprises, at least. Or, perhaps I'll take a part of Bob's policy and compose that with my own.

Bryan Ford: It seems likely that what's going to happen in that case is the only secure way that two policies that can compose is for each component to shut down the other's component and then nothing works.

Reply: Yes, and that's a fair point. But, if the composition of two objectives gives you 'bottom', the most restrictive policy, then that should indicate that the objectives need to be checked. At least I now have an algebra where I can compare policies and discover that there might be a problem, which is better than what we had before.

Bryan Ford: Yes. I guess this is the question. As attractive as an ordering is from a formal sense, is that practical as an answer for an average Joe to reason about security?

Reply: I'll answer using the theme of the presentation, which is about comparison. I believe that its better than what was there.

Paul Wernick: How do you address the situation where a change makes one part of the apparatus more secure, and one part a bit less secure?

Reply: That's a good question, and it is related to the earlier question about emergence. To deal with it you need to consider how the security objectives are upheld by the system components and security controls, and how these components behave and interoperate. For now, I've focussed on comparing just the objectives. It will be interesting to see how we might extend it to include the behaviour of the security controls.