SCADA: Publicly exposing obscure security flaws

Jeppe Rishede Thomsen
Department of Computing
Hong Kong Polytechnic University
Kowloon, Hong Kong
csjrthomsen@comp.polyu.edu.hk

1 Introduction

Computer security is a field which focuses on protecting IT assets - software and hardware - from natural and man made threats, such as e.g. viruses, theft, or water damage from excessive rain [13]. Security analysis is a sub-field of Computer security which is becoming ever more important as criminals are developing better and smarter techniques to break into, and take advantage of, computer systems [8]. Security analysis is the practice of analyzing a program, or parts there of, to find problems which might be exploited to gain unauthorized access to a system, like e.g. taking over a user account or being able to see restricted information.

When it comes to utilize security flaws and vulnerabilities, then Spam mail is still the preferred attack vector to gain some degree of control of a system [12]. Often security flaws are utilized via a small program, i.e. a virus/trojan/spyware, though the criminals behind these are increasingly starting to use a more direct approach with so called *trojanmware* where the victims system is being held "hostage" in some way by the criminals in order to extort money from the owners of the system. This can e.g. be achieved by encrypting some important data on the victims system, and not hand over the encryption key until payment has been made.

When it comes to industrial software systems as SCADA (supervisory control and data acquisition) which does not directly run either windows or Linux, then they have largely been ignored by criminals as there has been no money to be made from attacking such systems. SCADA systems have historically not been connected with regular regular networks, and the Internet, and they have therefor not been designed with security in mind [5]. SCADA systems are employed in places such as Oil refineries, nuclear plants, power plants, water treatment plants, etc. Such systems are designed to be data driven and the basic way they work has not changed for a long time, meaning that they adopt security by obscurity since the attacker would historically need special hardware and very specific knowledge, as well as physical access to attack such a system. These assumptions are however no longer true, often SCADA systems are connected to the Internet via the regular network on site, meaning a virus to a nuclear control system can be introduced by a normal employee connecting his laptop/phone/usb to the system. [4, 7]. It is not easy to fix the problems with SCADA, since the security problems are not bugs, but design decisions. It is not possible to introduce a user name/password scheme in most cases since SCADA compatible hardware mainly only supports anonymous login, or is supposed to run unsupervised for decades without changing the password. SCADA compatible hardware are e.g. water pumps, valves, or switches with can report their current state, and receive commands to change their state.

An, often overlooked, problem is the security researchers them selves. If a security researcher publishes a vulnerability that has not, or can not, be fixed, the he has ostensibly just given criminals and young curious hackers another way to attack an unpatched systems, which they did not have before. [1–3, 6, 7, 9–11]. While SCADA systems have been a potential target for government funded hackers for a while, then the "normal" hacker has simply not been aware of the existence of these systems before, but with the release of the stuxnet worm [4] people have become aware of the existence of SCADA systems.

2 The Case

I will show the ethical problems security researchers have to consider when researching and publishing vulnerabilities in SCADA systems. I will use the case of a well know SCADA systems researcher, Luigi Auriemma, and his insistence on always publishing his findings directly on the Internet.

Back in March 2011, Luigi Auriemma released 34 zero-day security issues[1] it is starting, and their proof of concept, for various SCADA control software. Luigi released the information without first notifying vendors and giving them a chance to fix the problems. Luigi maintains the position that it is always the best option to release vulnerabilities directly to the public, even thought the security issues then might be exploited in the window of time between publishing and a patch being released. In the case of SCADA security holes published, it might be even worse as some of the problems can not be fixed at all, since they are deeply embedded design decisions and not flaws in the systems.

[1] A zero-day security hole is a security hole which can are exploited on day one of the developers starting to fix the holes

Publishing security holes directly to a public bug tracker will guarantee that the vendors can not ignore and postpone releasing system patches, and in some cases it can be a necessary tactic in some cases when a vendor does not want to patch his system. There is however a clear ethical problem in using this as the default method of publishing security holes in software, as the security researcher will be exposing unpatched system to danger in the period between the publishing of the security holes, till a patch is released. Luigi is also knowingly publishing security flaws which he knows can not be fixed in the short term, thereby exposing critical infrastructure to danger, not just from the regular threat of terrorists (It has been show that Al-Qaeda had SCADA manuals [5]), but also regular script kiddies who just think it would be fun to see if they can access the local dam or power plant, possibly causing a lot of damage because they don't know the systems.

Luigi could obviously easily avoid exposing SCADA systems to unnecessary risks by simply working with SCADA system vendors and releasing any information about security risks to the vendors first. He could then release his findings after giving the vendors reasonable time to fix the security problems (it usually takes minimum a month to probably deploy a patch [5]). When it comes to the the inherent security problems of SCADA, which Luigi exposed, then it is obvious that these problems would be exposed at some point any way, as security by obscurity always will fail in the end. Despite this, then he should simply just have handed the information over to the vendors, and give them a head start on fixing the problems. This solution is a bit extreme, as it is in most cases in the publics interest that security issues are published. The problem here is the fact that SCADA systems are responsible for life supporting infrastructure so a SCADA specific virus wont just potentially mean lost income to some company because the machines did not work. A SCADA virus might mean loss of lives and entire countries without power for extended periods. One example could be if a dam got infected and just released all the water it held back, this would surely cause a lot of destruction and potential loss of lives.

3 Conclusion

Luigi Auriemma continues to do security research on SCADA systems and publishes his results directly to the public via bug trackers [2,3] and continues to claim that it is the right thing to to do.

There are other SCADA security researchers too out there, and the problems with SCADA are becoming more and more public, like e.g. Stuxnet [4] (attacked material to enrich uranium) which was in the media world wide. Since SCADA software manages such critical systems and information about SCADA manuals are readily available, we will likely see more viruses targeting SCADA systems. The unethical behavior of a few individuals, like Luigi, will probably not change much in the long run, but their actions will probably mean that we will see more SCADA exploits much sooner.

References

[1] R. A. How not to publish scada security advisories. March 2011.

[2] L. Auriemma. Re: Vulnerabilities in some scada server softwares. March 2011.

[3] L. Auriemma. Vulnerabilities in some scada server softwares. March 2011.

[4] BBC. Stuxnet 'hit' iran nuclear plans. Nov 2010.

[5] Blackhat.com. Scada security and terroism: We're not crying wolf.

[6] E. Byres. The italian job âĂŞ multiple scada/ics vulnerabilities go public. March 2011.

[7] J. E Dunn. Researcher uncovers more scada zero-day flaws. September 2011.

[8] R. King. Cisco: Cyber criminals keen on mobile device, cloud hacking. December 2011.

[9] A. Moscaritolo. Researcher again discloses multiple scada flaws. March 2011.

[10] v3.co.uk. New zero-day scada flaws raise spectre of stuxnet. March 2011.

[11] E. West. Security researcher finds, publishes scada vulnerabilities. September 2011.

[12] Wikipedia. 11 ways computer viruses are spread. 2010.

[13] Wikipedia. Computer security. January 2012.