Mitigating Local Attacks Against a City Traffic Controller

Nils Ulltveit-Moe[1], Steffen Pfrang[2], László Erdödi[1] and Héctor Nebot[3]
[1]*Faculty of Engineering and Science, University of Agder, Jon Lilletuns vei 9, 4879 Grimstad, Norway*
[2]*Fraunhofer IOSB, Karlsruhe, Germany*
[3]*ETRA I+D, Tres Forques 147, 46014 Valencia, Spain*

Keywords: Traffic Controller, Traffic Safety, Attack Mitigation, Critical Infrastructure.

Abstract: This paper demonstrates how a local attack against a city traffic controller located in a public area can be detected and mitigated in a cost-effective way. This is done by applying a general security methodology, an architecture and a set of new and existing tools integrated by the PRECYSE EU-project. The traffic controller does not contain built-in security and is connected to an information panel which is used for displaying traffic messages. The proposed solution is integrated with the incident management system of the city traffic control centre. This allows for increasing the situation awareness about attacks, as well as supporting a workflow for restoring the attacked device to its normal state and ensuring attack investigation.

1 INTRODUCTION

Traffic controllers are used for controlling information on road signs, traffic lights and similar, as well as for reading sensor information about number of cars, car speed etc. The safety critical nature of these technologies requires that such systems should be secured against cyber-attacks, however recent research has shown that this is not always the case (Ghena et al., 2014).

The technology used to implement these control systems is in many respects similar to SCADA systems. It is frequently based on old, expensive and obsolete technologies, and may have been implemented and designed for running in a trusted network with no extra security functionality installed. The systems work and are robust enough in normal operation, but have not been designed to resist or handle cyber-attacks.

Generalising the problem, this covers an important class of attacks on legacy software systems put outside the physical perimeter of an organisation which neither supports proper built-in security nor proper security analysis. These are typically widely distributed embedded systems that were installed a long time ago (e.g. traffic controllers, SCADA systems and similar) where security was not a main consideration during original deployment.

The approach described here is also useful for low-cost, widely distributed systems such as smart-grid Demand/Response systems. Adding support for security analysis to these systems is a good and cost-effective way of increasing the security awareness in case of malicious events, especially in less critical parts of an infrastructure, where some downtime of the monitored service can be tolerated and also where the infrastructure owners do not want to prioritise investing too much in more secure alternatives.

The PRECYSE EU project[1] investigated two use cases related to this: securing a traffic control network, which is described in this paper, and securing a Remote Terminal Unit (RTU) in the electricity grid against malformed IEC 60870-5-104 telecontrol traffic towards the SCADA server (Yang et al., 2013). Both these use cases utilise the general security architecture described in this paper, however the approach for detecting the attack is different. The traffic control network in this paper uses the OSSEC host-based intrusion detection system for detecting attacks, whereas telecontrol anomalies in the other use case was detected using a purpose-built Snort intrusion detection system rule set.

The security architecture is also being used for protecting the home energy management gateway and virtual power plant in a Smart-grid Demand Response system (Gjøsæter et al., 2014). This gives security awareness and some attack protection capabilities to a

[1]PRECYSE - Prevention and Reaction to Cyber-attacks to Critical Infrastructures: http://www.precyse.eu

Ulltveit-Moe, N., Pfrang, S., Erdödi, L. and Nebot, H.
Mitigating Local Attacks Against a City Traffic Controller.
DOI: 10.5220/0005649802090218
In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pages 209-218
ISBN: 978-989-758-167-0

low-cost embedded device performing load management of heaters and water boilers in a Zigbee based personal area network (PAN) in the home.

Another area where such technologies may be useful is for providing security awareness in home-based e-health systems, which often use a similar technology based around a gateway and a personal area network. The security architecture may also be useful for protecting rural critical infrastructures such as protecting the sensor network of municipal drinking and waste water processing as well as small power plants in rural areas in an upcoming project.

This paper aims at securing a traffic controller based on the venerable MS-DOS operating system, with a proprietary real-time extension running on top of it. MS-DOS pre-dated the widespread commercial use of TCP/IP, and does not have TCP/IP built-in to the operating system. It is still possible to run TCP/IP clients using MS-DOS, but this requires packet drivers and application layer protocol stacks to be added on top of this simple single-tasking operating system (Ozimek, 1996). Proprietary real-time kernels have however been developed on top of MS-DOS, using the Terminate and Stay Resident extension facility for adding simple multitasking. This allows controlling embedded devices, such as Programmable Logic Controllers (PLC) (Ramon Barth, 2011). Perhaps surprisingly, this is a technology that is still alive and is being deployed, maintained and extended in certain embedded systems, mainly because it does a simple job well, and there is no or little business incentive to port the legacy control software to a more modern platform.

One interesting finding when attempting to create an attack against this platform, was that the technology was so rare that the vulnerability detection system OpenVAS[2] did not detect any vulnerabilities against it. This does of course not mean that the software does not contain any vulnerabilities, only that potential vulnerabilities would require a significant amount of knowledge and effort to be exploited, because the traffic controller uses a proprietary control protocol with undisclosed proprietary control software. It may be possible to launch a man in the middle attack, for example using ARP spoofing to analyse the protocol which went in clear-text, and attack it using a fuzzer to identify possible vulnerabilities or modify the transmitted data from both directions. However, developing such attack scenarios was considered out of scope for our project, which focused on preventive security measures against local attacks. The main concern was therefore that attackers would break into the cabinet of the traffic controller, and connect a PC to it using a serial connection. Our objective was to identify this abuse case, and provide situation awareness to the traffic control centre in case an attack was launched against the controller.

Situation awareness here means notifying the operator in the City Traffic Control Centre via the Incident Management System (IMS) that the traffic controller had been tampered with, and then deploy countermeasures in order to restore the controller back to normal operation. The countermeasures were based on a set of existing and new security tools, which had been developed and integrated by the PRECYSE EU-project.

This paper is organised as follows: The next section describes the system architecture of the traffic control centre test-bed with added security domains. Section 3 describes the data flow for the intrusion detection and alerting system. Section 4 describes the attack case study, and how the tools could be used to detect and mitigate the attack. Section 5 does a security analysis of the attack scenario and section 6 does a more general discussion of advantages and limitations with the proposed approach. Section 7 covers related works, section 8 concludes the paper and section 9 outlines future work.

2 THE ARCHITECTURE

The security architecture (see Figure 1) integrates a set of Open Source tools and standards in order to increase interoperability, ease deployment and reduce costs (Kippe, 2014). It is based on the following top level components: An Enclave, which is a part of the target network, is under the control of a single authority and security policy. It forms its own security zone (Rome, 2012), and a set of security domains. Each security domain corresponds to a target enclave and comprises all enclave specific data acquisition and data processing. The domain consists of a set of security tools for protecting the enclave: a firewall (Shorewall[3]) for enforcing network segregation according to the security policy; a network based intrusion detection system (Snort NIDS[4]) for detecting security attacks on the enclave; a host-based intrusion detection system (OSSEC[5]) which can perform log analysis on behalf of the tools in the enclave; and a vulnerability tester (OpenVAS[6]) which is used for security scanning of the target enclave in order to detect

[2]OpenVAS http://www.openvas.org

[3]Shoreline firewall (Shorewall): http://shorewall.net

[4]Snort: https://www.snort.org

[5]OSSEC HIDS: http://ossec.github.io

[6]Open vulnerability assessment system (OpenVAS): http://www.openvas.org

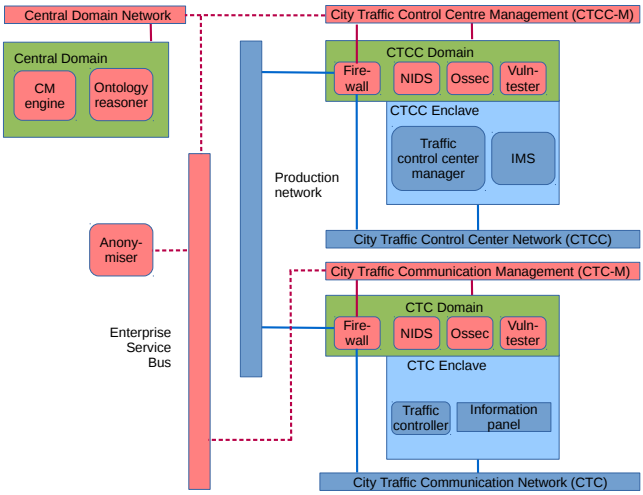

Figure 1: Simplified Architecture of Traffic Control Centre Testbed.

known vulnerabilities.

There are two enclaves in the test network. The first enclave represents the City Traffic Control Centre (CTCC) enclave, which contains the traffic control centre manager as well as the incident management system (IMS). The IMS is used to log traffic incidents on the second enclave: the City Traffic Communication Network (CTC), as well as security incidents detected by the security domains. There are furthermore two security domains in the test setup protecting each enclave: the CTCC Domain protects the city traffic control centre; and the CTC Domain protects the city traffic communication network, which runs the city traffic controllers that manage traffic information panels, traffic lights etc. There is a segregated management network for each domain, which is used for managing the security configuration using the network configuration protocol (NETCONF) (Enns et al., 2011), as well as conveying intrusion detection system (IDS) alarms in the Intrusion Detection Message Exchange (IDMEF) format (H. Debar, 2007). The city traffic controller management network (CTCC-M) is the segregated management network for the CTCC, and the segregated city traffic communication management network (CTC-M) protects the CTC.

The security tools are connected to the Central Domain, which performs security analysis and alert correlation using the Ontology reasoner based on Apache Jena. The ontology consists of four models: alarm ontology, attack ontology, system ontology and vulnerability ontology, which are used to perform event correlation and inference based on incoming events (Thomalla, 2014). The Ontology reasoner can forward alerts to the countermeasure (CM) engine, which can be configured to send requests for manual countermeasures to the IMS as well as automatic deployment of countermeasures (e.g. firewall updates) via the Domains based on the Common Remediation Enumeration (CRE) format (McGuire et al., 2011).

The services provided by the top level components are implemented as Web Services using the Enterprise Service Bus (ESB) Apache Servicemix as a message-oriented middleware thus providing network segregation and out-of-band management. This allows for flexibility in routing messages between different components according to the security policies of the network, including optionally running the messages through a reversible anonymiser which allows for fine-grained access control, anonymisation and deanonymisation of the information in these messages to different users or roles according to their security clearance and need (Ulltveit-Moe and Oleshchuk, 2012; Ulltveit-Moe and Oleshchuk, 2015).

3 DATA FLOW STEPS

The attack team developed a use case scenario for defining attack, detection and countermeasure steps. The attack and countermeasure method was being run in a controlled testbed, to avoid any safety issues that might occur by testing out these procedures in a live environment. In order to perform the attack detection, we integrated the chain of protection tools shown in Fig. 2.

3.1 Traffic Controller

The Traffic Controller (TC) is able to log events to a local log file. Among these, there are periodic events

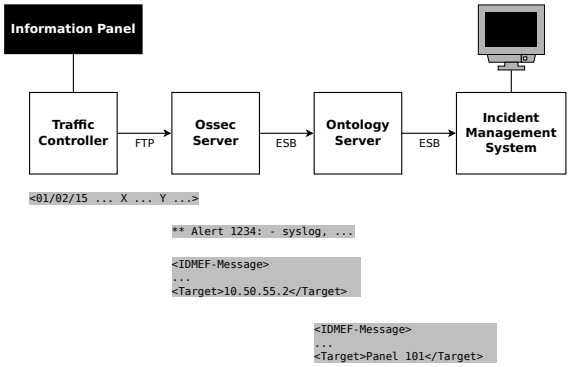

Figure 2: The data flow from the Information Panel to the Incident Management System.

and extraordinary events. Periodic events consist of values of measurements being performed on a regular basis, e.g. specific voltages. Extraordinary events are being logged on single events like “door opened”, “door closed”, “message changed”. Since the TC runs on the single task operating system MS DOS and does not support standards for log forwarding like Syslog, we had to implement a workaround with a local FTP server that makes the local log file available for remote access.

3.2 OSSEC Server

By default, the open source host-based intrusion detection system OSSEC supports the analysis of log files to remote systems by the means of an OSSEC agent. This agent has to be installed on the target system and communicates with the OSSEC server. A second possibility is the use of log forwarding with Syslog. Both solutions were not applicable to the scenario of the TC since they are not available for the legacy operating system. In order to overcome this issue, we made use of a cron job that retrieves the log file of the TC each minute via the File Transfer Protocol (FTP) and stores it locally on the OSSEC server. Then, we configured the OSSEC server, to treat the TC’s log like a local log file.

The format of the TC’s log file is proprietary and consists of plain text with one log item per row. In order to make OSSEC aware of this type of log file, we had to create a set of decoders (see Figure 3) that extract time, type of message and message from a given log item. In a second step, we introduced new rules to the OSSEC rule set that drop logs for regular actions and generate alerts if something goes wrong. Figure 4 depicts two rules that are being used to recognize if a new message has been presented at the Information panel.

```
<decoder name="etra-message">
 <parent>etra</parent>
 <prematch offset="after_parent">
   ^\sSMV
 </prematch>
 <regex offset="after_parent">
   ^\s(SMV)\W PAN(\d+) (\w+) mensaje
 </regex>
 <order>action,dstip,status</order>
</decoder>
```

Figure 3: Ossec decoder for log messages from the Traffic Controller.

```
<rule id="400012" level="3">
  <decoded_as>etra</decoded_as>
  <action>SMV</action>
  <description>
    Unknown message alarm
  </description>
</rule>
...
<rule id="900302" level="3">
  <if_sid>400012</if_sid>
  <status>Presenta</status>
  <description>
    Local message presented at panel
  </description>
  <info>
    A command to present a message in the
    information panel has been sent
  </info>
</rule>
```

Figure 4: Example of OSSEC rules that are being triggered when a new message is presented at an Information Panel.

3.2.1 IDMEF Creation

The OSSEC server processes log messages and generates alerts if specific rules are being triggered. These alerts are being stored in a local file using the native OSSEC alerts log file format. Internally, the IDMEF alert format was used for further processing, as shown in Figure 5 (H. Debar, 2007). The native OSSEC alerts log file format was converted into the IDMEF XML format. Additionally, each IDMEF formatted alert is made available for the central components via the ESB.

3.3 Ontology Reasoner

The OSSEC reasoner is aware only of IP devices. Therefore, the IDMEF messages contain information about involved systems being described by their IP address (e.g. 10.50.55.2). On the other hand, the Incident Mangement System (IMS) of the TC uses its own identifiers of devices, e.g. Panel1. In order to integrate the IMS, an enrichment of the IDMEF alert messages was done within the Ontology server, which

```
<IDMEF-Message>
  <Alert messageid="1423565102.8135">
    <Analyzer name="OSSEC"
     analyzerid="Ossec-CTC.vlc"/>
    <CreateTime ntpstamp="0xd8845fae.0x0">
      2015-02-10T10:45:02Z
    </CreateTime>
    <Target>
      <Node category="unknown">
        <Address category="ipv4-addr">
          <address>10.50.55.2</address>
        </Address>
      </Node>
    </Target>
    <Classification
     text="Local message presented at panel">
      <Reference origin="vendor-specific">
        <name>Rule:900302</name>
        <url>http://www.ossec.net/wiki/
             Rule:900302</url>
      </Reference>
    </Classification>
    <AdditionalData type="string"
     meaning="Source file">
      <string>
        /home/logfile/traces.txt
      </string>
    </AdditionalData>
    <AdditionalData type="string"
     meaning="Full Log">
      <string>
        &lt;10/02/2015 11:44:07.919 SMV&gt;
        PAN1 Presenta mensaje 2
        en el modulo 0
      </string>
    </AdditionalData>
  </Alert>
</IDMEF-Message>
```

Figure 5: Example of an IDMEF message.

is aware of the topology of the network and the mapping of a given IP address to the corresponding Panel id. This is a fairly simple approach that covers the needs of the local attack scenario under our given assumptions. Making more elaborate inferences based on correlation of IDS alarms is possible using the Ontology reasoner, however this is left as future research.

3.4 Incident Management System (IMS)

In order to allow the IMS to receive IDMEF formatted events from the ESB, a connector that subscribes to a specified topic in the ESB was implemented which receives these messages. The next step is parsing the relevant information inside the IDMEF messages and storing them in the SQL Server Database in order to feed the IMS with new incidents.

3.4.1 Enforcing Countermeasures

The IMS follows a predefined work-flow on reception of the IDMEF event, as illustrated in Figure 8: open

Figure 6: Traffic panel being hacked.

panel application and check current information presented in the panel; restore preprogrammed message from panel; call head of maintenance and close incident. This approach allows for defining consistent responses to different threat types according to need.

This approach works under the assumption that the traffic control panel is being managed completely by the traffic control system, so that local changes of the panel are not allowed.

4 ATTACK CASE STUDY

The attack scenario assumes a local attacker that breaks into a traffic controller and connects to it using Telnet, in order to perform commands for altering the content of the traffic panel. After the attacker has performed the attack, the panel shows the illegitimate panel message reproduced in Figure 6. It is here assumed that the attacker knows the Telnet password to the traffic controller.

All changes to the traffic panel are logged, and these logs are synchronised at regular intervals from the traffic controller and to the host-based intrusion detection system OSSEC in the CTC domain. The traffic controller will also log other events, such as the door to the controller being opened. OSSEC is configured to raise an alert if the door has been opened as well as if the panel message has been altered. It will forward this information via the central ESB to the Ontology-based correlation engine in the Central Domain. The correlation engine translates the source of the alert from an IP view to the corresponding naming scheme of the Traffic Management System and resends it to the ESB. Since the Incident Management System has subscribed to the specific alert topics in the ESB, the alert will appear in the GUI of the local operator.

The traffic control centre operator can then investigate the attack using the traffic management system, as shown in Figure 8. Since there has been an unauthorised change of the traffic panel, the operator can call the police to check out the incident with the panel and also restore the panel to its original state, as shown in Figures 7. This works under the assumption that the traffic panel only is being managed by the traffic control centre. Local modifications should not occur in production, and the Ontology reasoner

Figure 7: Traffic panel restored to original state.

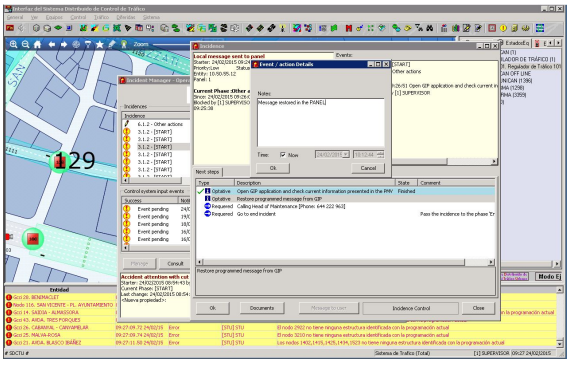

Figure 8: Intrusion alert detected.

will be able to infer whether the panel is in production or not. Even if the traffic panel in this case used an obsolete technology (MS-DOS), such a local attack may even be possible if a modern embedded operating system had been used, unless the device had taken special precautions, such as removing support for external storage devices, disabling the possibility to boot other images than the default image etc.

5 EVALUATION OF THE DEVELOPED SYSTEM FROM THE ATTACK TYPE POINT OF VIEW

It is obvious that the main propose of the attacker is to compromise the system. This can be achieved in four different levels as follows:

- Level 1: Compromise the traffic controller quickly without disguising the instant malicious activity. In this level the attacker does not care about the alarm messages that the traffic controller may send. The attacker takes advantage of the time factor. Even if the traffic control centre is probably informed about the attack, it takes time to intervene (police arrival, restoring to valid state). This type of attack is only good for causing relatively short chaos by disturbing messages.

- Level 2: Compromise the traffic controller and trying to disguise the instant malicious activity from the traffic control centre as long as it is possible. In this case the attacker has to stop the alarm messages from the local traffic controller and provide fake information to deceive the IDS of the system. This type of attack causes longer breakdown of the service thus it has a higher effect.

- Level 3: Compromise the traffic controller not instantly but with a plan of an attack in the near future. In this level, to be able to execute the attack, the attacker penetrates into the system without any instantly detectable signs and gains access to the system. This type of attack is perfect for an aligned attack in the near future if more than one traffic controller is under control by the attacker. Such attacks can have serious consequences since the number and location of the attacker-controlled traffic lights can be arbitrary.

- Level 4: Compromise the the traffic control centre itself using one traffic controller or maybe in another way (e.g. direct access to the traffic controller centre). In this rather unrealistic level the target of the attack is the traffic control centre directly. By breaking into the system of the traffic control centre, it would be possible for the attacker to send out compromised messages to all traffic controllers.

In all four levels the attacker can utilise the following approach: exploit unencrypted messages between the traffic controllers and the traffic control centre on the unencrypted LAN within the traffic controller.

It is assumed that traffic in and out of the traffic controller is controlled using a router with support for encryption, so that a local attack/physical break in to the traffic controller is necessary to get access to the more vulnerable internal LAN where the traffic goes unencrypted. This means that it is assumed necessary to physically break in to the traffic controller in order to carry out attacks. If the attacker breaks in to the TC, then the vulnerable unencrypted internal LAN connecting the traffic controller to the VPN router will be exposed. The attacker can then apply a ”man in the middle attack” via this LAN and obtain and/or modify the messages between the controller and the centre. In this case the attacker intercepts messages to get the physical address (MAC) of the communicating devices and generates fake messages with the spoofed MAC. This case can be applicable for attack levels 1-3. It can be mitigated by encrypting messages but this is out of the scope of the present paper.

In this case a level 1 attack is possible, since the attacker does not care about logging and alarm messages etc. In level 2,3 and 4 attacks the attacker has to be very quick stopping the sending of the suspicious logs to the traffic controller. One of the aims of our solution is to prevent these cases.

Since the developed system has a quick log message forwarding (in every minute the logs are forwarded to the traffic control centre using FTP), according to our analysis the attacker will not have enough time to carry out level 2,3 and 4 attacks just

Table 1: Security effect of our solution.

	without security	with proposed security
Attack level 1	possible	possible, but logged
Attack level 2	possible	protected
Attack level 3	possible	protected
Attack level 4	depends on the centre system	depends on the center system, but not through the controller

by breaking physically into the traffic controller, since the traffic controller already will have sent an alarm that the door to the TC has been opened. Table 1 summarizes the effect of our solution.

6 DISCUSSION

It must be noted that protecting against advanced persistent attacks on the proprietary TCP/IP based control protocol for the MS-DOS based traffic controller would require potentially very large effort, since it lacks support for cryptographic link protection, such as TLS or IPsec. This is perhaps one of the main risks of using an obsolete platform such as MS-DOS. However local attacks would be difficult to protect against, even with a more modern operating system, since such an attack easily can compromise the local disk to identify or modify user credentials etc. It is however assumed that this problem can be overcome for external attackers by building in a router with support for strong encryption into the traffic controller. This means that a local attack will be needed to compromise the traffic controller. Such a strategy could make sense from a business perspective, since a router with support for encryption can be built from common off-the-shelf components which are relatively cheap, whereas the traffic controller is a purpose-built system which typically is expensive to replace.

One could envisage that the protection mechanism perhaps could be attempted bypassed, for example by inserting a gateway intercepting the traffic between the traffic controller and the switch/router to the traffic control centre. Such an attack would however probably need the knowledge of an insider to be carried out successfully, since even a temporary change of MAC address for the traffic controller would be detected by the network-based intrusion detection systems (NIDS) in our test scenario, as shown in Fig. 1.

However if an attacker planned the attack well, then it might be possible for the attacker (for example an insider) to have measured the MAC address of the traffic controller, as well as typical traffic messages from it. It might then be possible for the attacker to write a traffic controller simulator, which exchanges a similar set of messages that would be indistinguishable from the original messages, including using the same MAC address. The attacker could then replace the real traffic controller with the traffic controller simulator, and modify the now offline traffic controller. This is assuming that the switch detecting "door open" signal could somehow be bypassed, for example by jamming the Internet connection. Our protection mechanism would not be able to detect such a scenario.

Other protection mechanisms would then be needed in order to detect the local attack, for example for example using a video camera monitoring the traffic controller. There are already several such cameras in use for monitoring the traffic, so this could have been an alternative way of detecting the attack, for example based on advanced video image analysis of the monitored traffic controller. We do however argue that the proposed approach is simpler and cheaper, with sufficient reliability in order to at least detect opportunistic tampering of the traffic controller as well as unauthorised changing of traffic messages by external attackers. If the service was considered critical enough, then additional protection could be set up, if a risk analysis shows that this would be necessary.

There is after all a limit for how much security investments a stakeholder is willing to do. It has been proposed that it does not make sense for a risk-neutral actor to invest more than a fraction of the expected losses (maximum 37%) due to security breaches (Gordon and Loeb, 2002), which probably is the reason why the traffic control centre initially considers implementing low-hanging fruits like this from a security perspective. The security losses due to such cyber-attacks have so far been low, and the main concern is that they currently are blind if such attacks occur, at the same time as they know that such attacks have been demonstrated. Furthermore, such a local attack on a traffic controller is a type of attack that does not easily scale, as opposed to a remotely exploitable attack. Also, the proposed mitigation raises the cost of attack for the attacker, since there will be a significant risk of being caught when the traffic controller signals an attack.

There are for example already surveillance cam-

eras in several crossings of the city which can be correlated with information from the traffic controller. This means that the risk of such attacks probably is smaller for traffic controllers than attacks on for example smart meters, where customers might have an economic incentive for fraud, as has been demonstrated in a set of Maltese fraud cases[7]. The currently known attacks on traffic controllers have been performed by security experts and "honest but curious" hackers. It would be unreasonably expensive to replace all city traffic controllers due to this risk, especially for public authorities that need money also for other good causes than traffic control. This means that increasing security awareness using intrusion detection systems is the most natural thing to start with from a business perspective, since the existing traffic controllers still work and do their job perfectly. There are however plans to upgrade to a newer technology, something that will be done gradually over time.

The decision to use open source security software may also be considered a risk, since this may introduce some well-known vulnerabilities into the system. Mitigating existing known vulnerabilities is handled in the security architecture using up-to-date security testing tools, such as OpenVAS, which are connected to the Verinice[8] information security management system via our own Asset Reporting Format (ARF) to Verinice .vna import filter *arftoverinice.py*[9]. The import filter contains a simplified threat model based on the CVSS score, which allows for performing a risk assessment based on identified vulnerabilities, in order to mitigate the vulnerabilities that are considered too risky.

The most secure method against local attacks, is probably utilising hardware supported security, for example by using a Trusted Platform Module, and using encryption, message authentication codes and sequence numbers to protect disks and communication protocols. This would however require replacing the infrastructure, something that should be done gradually according to need. Some additional security can be achieved by disabling local ports, however this might conflict with requirements for maintainability.

The chosen protection strategy increases the situation awareness of local attacks against the traffic controller, however even this strategy is not completely secure against attacks, since it could be possible to mount man in the middle attacks, for example using ARP spoofing on the internal LAN of the traffic controller, which could stop the log messages. However, such an attack will be detected by the tool chain since the ARPWATCH[10] tool is in place that alerts on a change in the assignment of MAC and IP addresses.

Additionally, it has to be noted that the chosen approach is highly flexible and transferable to other kinds of critical systems. Even if this traffic controller use case has been tailored to protect a legacy system, the general system architecture will be the same when dealing with a modern system (Yang et al., 2013; Gjøsæter et al., 2014). The only precondition is that the application can be configured to produce log messages which can be processed by the OSSEC Server.

In general, the chosen approach increases the situation awareness significantly and makes it harder for an attack to go undetected. However, a limitation is that the approach is not able to avoid a successful attack, since the attacker is able to change the road sign for a small period (some minutes) until it has been successfully restored by the traffic control centre.

The evaluation that has been done of the system, is a relatively simple case study. It is envisaged that more comprehensive studies of the security architecture will be needed in the future. However, the proposed solution solves an immediate problem that the traffic control centre recognises - the reputational risk in case the traffic controllers are hacked. This is important, since such events have got quite much press in the past (Ghena et al., 2014).

7 RELATED WORKS

Our paper describes an approach for increasing the security awareness of attacks on a city traffic controller. The idea of applying such measures on a traffic controller without built-in security is to the best of our knowledge not described elsewhere. The general system architecture used for implementing the protection measures has also been demonstrated for protecting SCADA traffic in energy systems (Yang et al., 2013), as well as proposed for protecting the security of Smart-grid Demand Response systems (Gjøsæter et al., 2014). This shows that the system architecture is relatively general with a broad application area also beyond protecting traffic control systems.

Other papers that have pointed out the lack of security in road traffic control systems, is for example (Ghena et al., 2014). This paper focuses on the wire-

[7]Malta's smart meter scandal: http://www.smartgridnews.com/story/maltas-smart-meter-scandal-41-million-worth-electricity-stolen/2014-02-18

[8]Verinice information security management system: http://verinice.org

[9]ARF to Verinice conversion filter http://sourceforge.net/projects/arftoverinice/

[10]ARPWATCH http://linux.die.net/man/8/arpwatch

less security in road traffic control systems, which often is poor or non-existing, for example relying on unencrypted wireless connections, default passwords that cannot be changed since management tools require them, no replay protection in the protocols etc. It demonstrates several working exploits against the traffic controller. Similar problems have been found with air traffic control systems, since the protocol is not being encrypted (Costin and Francillon, 2012).

A general problem pointed out in these papers, is the lack of security consciousness in this field. Attacks on traffic control systems via the wireless interface is also demonstrated in (Cerrudo, 2014). Both of these papers propose the use of good practices for protecting traffic controllers, but they do not demonstrate a practical solution for detecting local attacks on a device with limited capability for protection. Our paper focuses on local attacks against the traffic controller, since this was considered one of the greatest risks for our use case. It is assumed that network security can be achieved by applying existing best practices, such as using using routers supporting IPsec encryption.

A distributed smart signal architecture for traffic signal controls has been proposed in (DeVoe and Wall, 2008). This architecture suggests using RSA encryption for security, but does not consider how to mitigate local attacks.

The paper is also somewhat related to critical infrastructure protection and protection of SCADA systems in general, for example (Cheung et al., 2007; Igure et al., 2006; Stouffer et al., 2006; Luallen, 2011). The strength with our solution is demonstrating how a vulnerable device with no built-in security can be protected against local attacks using surrounding security services.

8 CONCLUSION

This paper demonstrates how attacks on a vulnerable city traffic controller can be detected and mitigated in a cost-effective way by applying the security methodology, architecture and tools developed by the PRECYSE project. The use case demonstrated in this paper shows how the tools are able to detect an on-going attack on the traffic controller, and automatically alert the operator in the control centre that is able to restore the traffic controller to its original state, as well as ensuring that the incident is being reported, in order to stop the on-going attack. Previous works have illustrated that this general architecture can be applied to enhance the security of SCADA systems in the energy sector, as well as protecting Smart-grid Demand Response systems. This means that the existing safety system has been enriched with security information, which is far better than current practices, where such attacks may go totally unnoticed. The architecture and tools are especially useful for protecting against attacks on legacy software systems put outside the physical perimeter of an organisation which neither supports proper built-in security nor proper security analysis. Adding attack detection and mitigation capabilities to such systems is very useful, even if the protection is not perfect, since it raises the perceived cost of attacks by increasing the risk of being detected.

9 FUTURE WORK

City traffic controllers should in the future embed improved security solutions for avoiding such local attacks. The traffic controllers should also embed more security in the control protocol, so that displayed messages at least would need to be authenticated using a cryptographic message authentication code in order to be displayed. This would make it far more difficult for the attacker to create unauthorised messages on the information panel.

Future work is also considering more advanced attack scenarios of traffic control systems, where the entire system could be modelled as a cyber-physical system, and anomalies in behaviour could be flagged up as alerts. This includes performing comparative evaluations and more experimental results.

ACKNOWLEDGEMENTS

Thanks to the anonymous reviewers, who gave good and challenging questions and suggestions for improving the paper. This paper has been developed as part of the FP7 EU project: PRECYSE - Protection, prevention and reaction to cyberattacks to critical infrastructures, contract number FP7-SEC-2012-1-285181 (http://www.precyse.eu).

REFERENCES

Cerrudo, C. (2014). Hacking US Traffic Control Systems. https://www.defcon.org/images/defcon-22/dc-22-presentations/Cerrudo/DEFCON-22-Cesar-Cerrudo-Hacking-Traffic-Control-Systems-UPDATED.pdf.

Cheung, S., Dutertre, B., Fong, M., Lindqvist, U., Skinner, K., and Valdes, A. (2007). Using Model-based Intrusion Detection for SCADA Networks. In *Proceedings of the SCADA security scientific symposium*, volume 46.

Costin, A. and Francillon, A. (2012). Ghost in the air (traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices. *Black Hat USA*.

DeVoe, D. and Wall, R. (2008). A distributed smart signal architecture for traffic signal controls. In *IEEE International Symposium on Industrial Electronics, 2008. ISIE 2008*, pages 2060–2065.

Enns, R., Bjorklund, M., Schoenwaelder, J., and Bierman, A. (2011). RFC 6241 Network Configuration Protocol (NETCONF).

Ghena, B., Beyer, W., Hillaker, A., Pevarnek, J., and Halderman, J. A. (2014). Green Lights Forever: Analyzing the Security of Traffic Infrastructure. In *Proceedings of the 8th USENIX Conference on Offensive Technologies*, WOOT'14, pages 7–7, Berkeley, CA, USA. USENIX Association.

Gjøsæter, T., Ulltveit-Moe, N., Kolhe, M. L., Jacobsen, R. H., and Ebeid, E. S. M. (2014). Security and Privacy in the SEMIAH Home Energy Management System.

Gordon, L. A. and Loeb, M. P. (2002). The economics of information security investment. *ACM Trans. Inf. Syst. Secur.*, 5(4):438–457.

H. Debar, D. Curry, B. F. (2007). *The Intrusion Detection Message Exchange Format (IDMEF)*. IETF.

Igure, V. M., Laughter, S. A., and Williams, R. D. (2006). Security issues in SCADA networks. *Computers & Security*, 25(7):498–506.

Kippe, J. (2014). Cyber-security in kritischen infrastrukturen. In *visIT IT-Sicherheit für die Produktion*, number ISSN 1616-8240 in 15. Fraunhofer IOSB.

Luallen, M. E. (2011). Critical Control System Vulnerabilities - And What to Do About Them. http://www.sans.org/reading-room/whitepapers/analyst/critical-control-system-vulnerabilities-demonstrated-about-35110.

McGuire, G. T., Waltermire, D., and Baker, J. O. (2011). Common Remediation Enumeration (CRE) Version 1.0 (Draft).

Ozimek, I. (1996). Accessing MS-DOS applications over a TCP/IP network. *Microprocessors and Microsystems*, 20(1):31–38.

Ramon Barth (2011). Real-time processing - the basis for PC Control.

Rome, J. A. (2012). Enclaves and Collaborative Domains. *BEYOND˜*, page 252.

Stouffer, K., Falco, J., and Kent, K. (2006). Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security. Technical Report NIST 800 82, NIST.

Thomalla, C. (2014). Ontologie-basierte erkennung. In *visIT IT-Sicherheit für die Produktion*, number ISSN 1616-8240 in 15. Fraunhofer IOSB.

Ulltveit-Moe, N. and Oleshchuk, V. (2012). Decision-cache based XACML authorisation and anonymisation for XML documents. *Comput. Stand. Interfaces*, 34(6):527–534.

Ulltveit-Moe, N. and Oleshchuk, V. (2015). A novel policy-driven reversible anonymisation scheme for xml-based services. *Information Systems*, 48:164 – 178.

Yang, Y., McLaughlin, K., Littler, T., Sezer, S., Pranggono, B., and Wang, H. (2013). Intrusion Detection System for IEC 60870-5-104 based SCADA networks. In *2013 IEEE Power and Energy Society General Meeting (PES)*, pages 1–5.