Author(s): Tenkanen, Tuomas; Hämäläinen, Timo

Title: Security Assessment of a Distributed, Modbus-based Building Automation System

Year: 2017

Version:

Please cite the original version:

Tenkanen, T., & Hämäläinen, T. (2017). Security Assessment of a Distributed, Modbus-based Building Automation System. In CIT 2017 : 17th IEEE International Conference on Computer and Information Technology (pp. 332-337). IEEE. https://doi.org/10.1109/CIT.2017.38

Security Assessment of a Distributed, Modbus-based Building Automation System

Tuomas Tenkanen and Timo Hamalainen
Faculty of Information Technology
University of Jyvaskyla
Jyvaskyla, Finland
Email: tuomas.s.tenkanen@jyu.fi, timo.t.hamalainen@jyu.fi

***Abstract*— Building automation systems were designed in an era when security was not a concern as the systems were closed from outside access. However, multiple benefits can be found in connecting such systems over the Internet and controlling a number of buildings from a single location. Security breaches towards building automation systems are increasing and may cause direct or indirect damages to the target organization or even the residents of the building. This work presents an approach to apply a method of data flow recognition and environment analysis to building automation through a case study on a distributed building automation system utilizing the Modbus protocol at the sites and presents suggested methods for mitigating the risks.**

Keywords— building automation systems; security; protocols

I. INTRODUCTION

Building automation systems have a long lifespan and the protocols used in existing, recent and even new installations were designed years or decades ago, in an era when security was not a concern as the systems were supposed to be closed. However, there are multiple benefits in connecting building automation systems so that connected buildings can be controlled and monitored from on place as changing a setting in the automation system does not require human intervention at the building site. As with any other connected system, also building automation systems may be attacked against for either direct or indirect benefit.

The distributed building automation system studied consists of a network of several dozens of buildings connected over a virtualized LAN to a data center and then to an office with the workstations where the buildings can be monitored and controlled. There are Modbus to IP converters at the buildings and the information to and from the buildings is transferred over IP. Thus, the network can be described as a fourth generation building automation and control system as described in [1]. Utilizing Internet, network virtualization and tunneled connections building automation systems can benefit from the concepts of IoT and thus the controlled buildings have virtual representations within the network as in [2].

The attacks against building automation systems are increasing [3] as the vulnerabilities regarding the systems are better known and they can be exploited regardless of time or place with little risk of getting caught [4]. Deeply integrated building automation systems may offer attackers and malware a significant attack vector both from the inside and outside of the organization [5]. Building automation systems have been utilized in large-scale attacks, with the Target breach in 2013 possibly being the best-known example of such an attack [6].

In this work, we apply a method of recognizing data flows of a distributed building automation system within the networks it passes, assess the vulnerabilities and risks of each network segment regarding the data separately and present an analysis of the case with suggested methods to mitigate the risks identified.

The paper is organized as follows. Section II discusses the related work concerning security audits and assessments in general and in the context of automation systems. Section III presents the methods used in the research. Section IV describes the network and the systems studied as well as the findings. Section V then contains the suggested methods and countermeasures to mitigate the risks identified in section IV. Lastly section VI concludes the results and presents the planned future research ideas.

II. RELATED WORK

Security audits and assessments have been studied at large. In [7] the process is described in general starting from recognizing vulnerabilities, assessing them and applying countermeasures to testing, monitoring and repeating the process. Technical methods are described in [8] as well as recognizing personnel as part of the system. Four levels of automation of a security audit are found in [9] and the importance of automating the audit process is discussed in [10] due to repeatability of the process and the large amounts of data. Various methods of security audits and assessments are presented in [11] and [12].

Information security in building automation systems has been studied in [4], [5], [13] and [14]. Demonstrations of working breaches have been shown in [15] with discussion over the attack motivations, methodology and possible damages as well as the difficulty of improving the situation quickly. Traditionally [16] building automation systems have not been connected over the Internet due to price of the

gateway devices and the poor interoperability of automation and TCP/IP protocols. However, the situation is changing and IPv6 could provide direct access to all such systems [2], which would accentuate the need for security even further.

The related works describe theoretical models or analyze high-profile breaches whereas our work analyzes an everyday building automation network utilizing the chosen model.

III. Data collection and manipulation

The chosen research method was the identification and modeling of data flows within complex networks and then the analysis of each separate network segment. Data flow analysis in various environments has been described by [1], [2], [3] and [4]. Data flows in building automation networks are utilized in [5] and [6]. The methods used to conduct this work are based on [7], however performed manually.

The process to identify and discover the data flows within the network segments includes the following six phases [7]:

A. Information collection

Common tools are used to map network connections, topologies, servers, and software. Detailed information is acquired by analyzing configuration files or code analysis. Collected data can be augmented with interviews. In this work information collection was done by interviewing the personnel of the organization managing the buildings, providing the connectivity between the buildings, running the data center and the office as well as the provider of the building automation system components. Technical methods to map the network and devices connected to it included using common port mapping and traffic capture tools both at the office where the building automation system is controlled as well as a few buildings.

B. Information filtering

Information is filtered with the aim of filtering out unneeded data arising from e.g. systems management software or network storage traffic. In this work, unnecessary data was filtered out by analyzing the network traffic captures performed at the office location. The captures done at the buildings only contained building automation data and some network virtualization data.

C. Data Flow Identification

Known traffic types can be classified and grouped by type. Unknown connections can be classified and later reviewed. For this study network topology maps were analyzed to identify data flows within the virtualized LANs between the buildings, the data center and the office.

D. Component Grouping

Similar data flow types can be clustered allowing easier representation. Other than the office with several usual data flow types this phase was not necessary as the network between the buildings and the data center is isolated from other networks by virtualization.

E. Data Flow Mapping

Place the data flows in the network topology allowing the recognition of network segments for further analysis. The building automation data was placed on the network topology recognized in the earlier phases.

F. Verification

Human knowledge confirmed the automated findings. Verification was done in interviews with the connection provider and the data center operator.

The parties involved in data transfer and storage provided high-level network topology maps for the research. Neither was willing to disclose details such as the physical location of the connections, routers or the servers, which of course was not necessary from the network point of view. However, it should be remembered that physical access to any network would allow access to data or at least the possibility to disrupt or deny the availability of the service. The connections from the buildings to the office and further to the data center and back were assumed safe and secure based on their descriptions and service agreements.

The party providing the components for building automation described their access to the network and the Modbus/IP converters. According to them they have access to the converters through a web interface, which is accessible over a commercial remote desktop client. They also provide access for maintenance and HVAC partners.

The high-level network topology maps were further brought into detail by performing network mapping and port scanning at the office and the building automation network. Both networks were first scanned with arp-scan and then selectively mapped with nmap. Also, the wireless LANs at the office were evaluated for security by injections and traffic capture.

Within the office network several hosts were found with open ports and services, especially http was commonly open. A selected set of the host with open services is presented in table I. One of the hosts was recognized as a security camera by the examination of the web interface and then scanned more thoroughly. Many ports were open on this host and the open ports are presented in table II. Another host with a completely different port profile was also found and then scanned. The results of this scan are in the table III. An interview with the office personnel revealed that this host was used to manage physical keys within the organization.

The building automation network was also scanned first with a ping scan and selected hosts were then scanned more thoroughly. Most of the devices were Modbus/IP converters as expected, which was confirmed by performing a scan against selected ports. No network devices other than the gateway were found revealing the fact that all buildings were connected to a single network segment. This was further confirmed by performing traceroute queries that showed that all buildings were one hop away from every other device. One of the converters at a building site was chosen for a complete TCP scan with operating system and service version fingerprint

identification enabled. This revealed some very old operating system kernel and service versions described in table IV.

TABLE I. RESULTS OF THE PORT SCAN IN THE OFFICE NETWORK

IP	TCP ports					UDP ports		Note
	22	*53*	*80*	*139*	*433*	*53*	*67*	
.1	open	close	open	close	close	close	o|f	1)
.5	open	close	open	close	close	close	close	2)
.6	close	filter	open	close	open	close	close	2)
.20	close	close	open	close	open	close	close	3)
.21	close	close	open	close	open	close	close	3)
.22	close	close	open	close	open	close	close	3)
.62	filter	filter	filter	filter	filter	o|f	o|f	
.63	close	close	close	open	close	close	close	
.64	close	close	open	filter	close	close	close	4)
.65								
.66	filter	filter	filter	filter	filter	o|f	o|f	
.67	filter	filter	filter	filter	filter	o|f	o|f	
.69	filter	filter	filter	filter	filter	close	o|f	
.70	filter	filter	filter	filter	filter	o|f	o|f	
.71	filter	filter	open	open	filter	o|f	o|f	
.72								
.73	filter	filter	filter	filter	filter	o|f	o|f	
.74	close	close	open	open	close	close	close	
.75	close	close	close	open	close	close	close	5)
.76	filter	filter	filter	filter	filter	o|f	o|f	
.81								

[a.] 1) router/gateway, 2) Wi-Fi base station, 3) network printer, 4) security camera, 5) VNC web UI

TABLE II. RESULTS OF THE PORT SCAN ON THE SECURITY CAMERA

PORT	STATE	SERVICE
25/TCP	open	smtp
80/TCP	open	http
135/TCP	open	msrpc
139/TCP	open	netbios-ssn
445/TCP	open	microsoft-ds
554/TCP	open	rtsp
2869/TCP	open	icslap
5357/TCP	open	wsdapi
8000/TCP	open	http-alt
8081/TCP	open	blackice-icecap
8082/TCP	open	blackice-alerts
10243/TCP	open	unknown
49152/TCP	open	unknown (upnp)
49153/TCP	open	unknown (upnp)
49154/TCP	open	unknown (upnp)
49155/TCP	open	unknown (upnp)

TABLE III. RESULTS OF THE PORT SCAN ON THE KEY MANAGEMENT WORKSTATION

PORT	STATE	SERVICE
135/TCP	open	msrpc
139/TCP	open	netbios-ssn
445/TCP	open	microsoft-ds
623/TCP	open	oob-ws-http
1342/TCP	open	unknown
1343/TCP	open	unknown
1344/TCP	open	unknown
1345/TCP	open	unknown
1346/TCP	open	alta-ana-lm
1347/TCP	open	bbn-mmc
1348/TCP	open	bbn-mmx
1349/TCP	open	sbook
1350/TCP	open	editbench
1351/TCP	open	equationbuilder
4082/TCP	open	unknown
4083/TCP	open	unknown
4084/TCP	open	unknown
4085/TCP	open	unknown
4086/TCP	open	unknown
4088/TCP	open	unknown
4089/TCP	open	unknown
4090/TCP	open	omasgport
4092/TCP	open	unknown
5623/TCP	open	unknown
5800/TCP	open	vnc-http
5900/TCP	open	vnc
6552/TCP	open	unknown
16992/TCP	open	amt-soap-http

Also, one Raspberry Pi computer was found from the building automation network with ssh and http ports open. Discussion revealed that this computer was used for network diagnostics earlier but was then forgotten.

At the building sites traffic was recorded for 24 hours. The capture was done with a Raspberry Pi 3 computer with an extra Ethernet adapter configured as a bridge and placed next to the Modbus/IP converter. The capture setup is pictured in figure 1 and the bridge device location is described in figure 2. The bridge device was invisible towards the network but remained accessible over the Wi-Fi hotspot configured on the Pi's Wi-Fi interface. Traffic received on either of the Ethernet interfaces was passed through the bridge device without alteration but also captured to files with tcpdump. The size of the data dump was 42 megabytes containing 449,580 packets.

TABLE IV. SOFTWARE VERSIONS IDENTIFIED IN THE MODBUS/IP CONVERTERS

SERVICE	SOFTWARE	VERSION	RELEASED
kernel	Linux	2.6.9-2.6.33	Oct 2004 – Feb 2010 [8]
ftp	unknown		
telnet	utelnetd	unknown	May 2002 – Aug 2008 [9]
http	Boa HTTPd	0.94.14rc20	Jun 2004 [10]

Fig. 1. Bridge device capturing traffic at a building

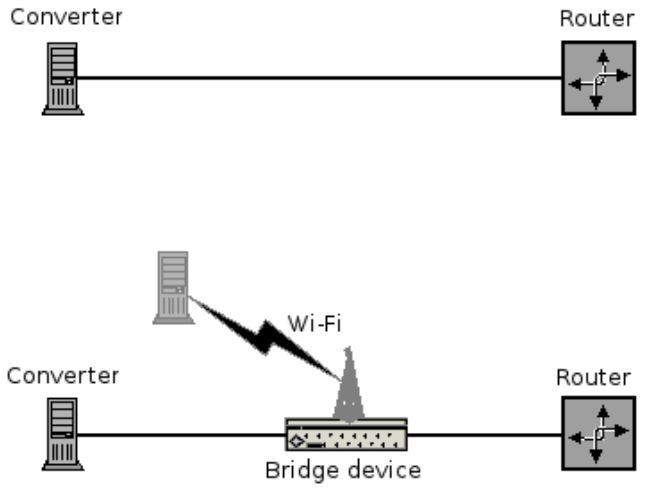

Fig. 2. Bridge device added between the Modbus/IP converted and the building automation network router.

IV. Structure of the Distributed Building Automation Network and the Findings

The building automation network consists of a virtual LAN connecting the building sites to each other and to a firewall/router device located at the office building. The data center is also connected to the same firewall/router device as well as the whole office network including the building automation system control workstation, regular workstations, Wi-Fi base stations and other devices. The network is described in figure 3.

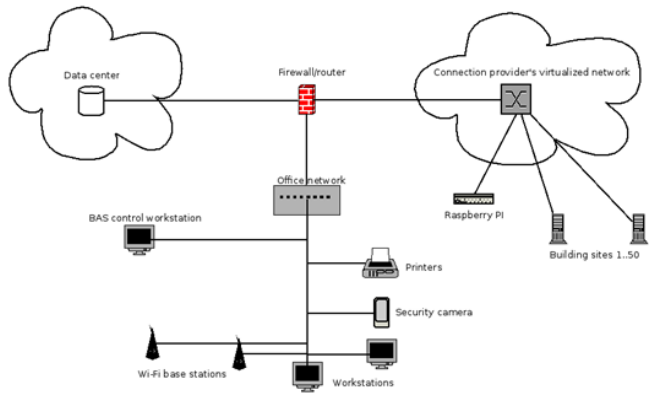

Fig. 3. Simplified network topology of the networks.

A. Known vulnerabilities of the identified software

Without direct access to the Modbus/IP converter the exact Linux kernel version running the device could not be identified. Based on the nmap operating system fingerprint identification the version was between 2.6.9 and 2.6.33. Linux kernel version 2.6.32 contains more than 180 known vulnerabilities [26] including at least two that allow a complete denial of service remotely [27].

BoaHTTPd development has ceased in 2005 [25] and contains at least one unpatched remote execution vulnerability [28] that is known to have been used in distributed denial of service attacks [29].

The key management workstation was running Windows XP, which is no longer supported by Microsoft [30] and has at least 45 known exploits [31].

B. Traffic capture analysis

The traffic capture at the controlled building shows a rather large proportion of ARP packets, which the network devices use to communicate the status of the network to each other, at 46 percent of the traffic. ARP traffic contained both the traffic of the connection provider's devices and the traffic of the building automation system converters. The ARP traffic revealed that only one device was located at the building site as expected. ARP traffic from one device also reveled a misconfiguration of said device as it probed for one IP address for more than 50,000 times in 24 hours. The IP address in question was not assigned to any device, but is a usual address of a gateway device.

The second largest share of packets was UDP traffic at 30.7 percent. The amount of UDP traffic was caused by the building automation system control messages that are used to sent commands and receive information between the building automation system and the control server located at the data center.

The last significant share of traffic was made ICMP packets. The devices pinging each other, especially the gateway, the Raspberry Pi and the converters at building sites, caused this.

A small amount of TCP traffic was captured that was apparently incorrectly routed based by the IP addresses within the captured packets.

C. Possible attack scenarios

In case an attacker has access to the building automation network in question, she may affect the operability of the system in multiple ways. It is possible to gain access to other connected systems [6]. Due to the structure of the network the attacker may proceed from one building to another freely as there is no segmentation within the building automation network. However, all these scenarios require physical access to the buildings or some part of the building automation network and thus thorough knowledge of the system and the network topology.

The Modbus/IP converters at the buildings are low-powered computers that can only handle a small amount of network traffic. The network is high-speed and thus can deliver large traffic and thus the easiest way to affect the operability of one or many converters is to overwhelm them with simple denial of service attack [32].

It is also possible to perform a denial of service attack on the side of the automation devices if the attacker has a

possibility to gain access to the devices. Such an attack has been described in [33] with suggested mitigation methods.

The remote vulnerabilities known in the Linux kernel in the Modbus/IP converter devices allow for a complete denial of service. Each of the converters in the building sites could be brought down with sending just one single packet to the broadcast address of the building automation network.

BoaHTTPd allows remote execution of any code on the device through a properly formatted HTTP request. The exact attack method and the usability of such an exploit depend on the software installed on the converter device.

With physical access to the building automation network the attacker could also perform more sophisticated attacks, such as modifying the data transmitted from the Modbus/IP converters. Modifying the data e.g. with placing a bridge device similar to one used in this research would be easy as the data is not encrypted and follows the Modbus protocol. Such an attack could have serious implications on the integrity of the building automation system data. Such attacks have not yet known to be done [33].

All the attack methods described before require physical access to the building automation network and deep knowledge of the devices therein. Because of this it would seem likely that other entry points to the network would be of interest to the attacker, such as mobile phones or laptops with remote desktop capabilities. This might allow access to the building automation server but not directly to the automation network.

Preventing physical access to a building automation network completely might not be possible as the residents and other users of the buildings need access close to e.g. HVAC and lighting systems [13]. It has been shown in [15] that access to a building automation network may be gained without actual physical access.

V. Suggested mitigation methods

In order to mitigate the possibilities to exploit the building automation system in question the following methods and procedures should be implemented:

A. *Update software where possible*

Some very old software versions were found both in the office and building automation network. Where possible, such old software should be updated. However, it is often not possible to update parts of e.g. Modbus/IP converters' software, but request a complete firmware update from the manufacturer.

B. *Security policy*

The building automation network operator did not have a written security policy in effect. Multiple members of the personnel however have access to the building automation system and e.g. various mobile devices that were allowed to use both at the office and at home. The policy should also be communicated and trained to the personnel and updated as necessary.

C. *Up-to-date documentation*

No one in the organizations involved had a complete picture of the building automation system as a whole. The system had several providers and each of these providers had versions of documentation of their supplied part, some up-to-date, some not. It is not possible to verify the state of any information system if there is no documentation of it. Thus a documentation of the whole system should be created and not just of its parts.

D. *Segmentation of the building automation network*

The end-points of the building automation network, either at the buildings or the office, were recognized as most potential entry points for an attacker. As such the network allows anyone with access to one entry point to operate freely with all devices within the network. Considering that the Modbus/IP converters are all very similar to each other, having breached or infected one, it is easily possible to breach or infect all the converters. Arranging the network to smaller segments would limit or slow down movement within the network.

E. *Technical monitoring of the automation network*

Currently there is no monitoring of the building automation network implemented. Building automation traffic is generally very well behaved as described by [34] and thus anomalies should be easily spotted. An intrusion detection system should be implemented to detect possible attacks [35].

F. *Device-level access control*

The building automation network converters are stable by nature and they are not added to or removed from the network frequently. Thus device-level access control by using MAC addresses should be used to allow only known devices to operate on the building automation network.

G. *Investigating unusual behavior*

One Raspberry Pi computer was connected to the building automation network without immediately known purpose. One Modbus/IP converter sent unusual ARP queries and some incorrectly routed traffic was captured within the building automation network. As the network should be very stable, all such events should trigger an investigation.

VI. Conclusion and Future Work

In this work we have shown that even a well-secured building automation network has vulnerabilities that are difficult or even impossible to patch without the manufacturers being active. We have also shown that an attacker must have thorough knowledge of computer vulnerabilities and exploits as well as automation systems in general and especially of the targeted system to be able to penetrate or breach a building automation system. As there is no practical way of completely preventing physical access to a building automation system, it is important to apply a defense-in-depth approach to the network to limit movement in case of a breach.

As future work, we would like to present a Proof of Concept attack on a Modbus/IP converter like the ones used in the building automation system described in this work. The Proof of Concept work could be done on individual software components should such a converter not be available.

REFERENCES

[1] J. Li, Y. Zhang, and F. Kuang, "Intelligent Building Automation and Control Based on IndasIBMS," in 2013 International Conference on Service Sciences (ICSS), 2013, pp. 266–270.

[2] M. Jung, C. Reinisch, and W. Kastner, "Integrating Building Automation Systems and IPv6 in the Internet of Things," in 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012, pp. 683–688.

[3] P. Čeleda, R. Krejčí, and V. Krmíček, "Flow-Based Security Issue Detection in Building Automation and Control Networks," in Information and Communication Technologies, vol. 7479, R. Szabó and A. Vidács, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 64–75.

[4] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks Against Process Control Systems: Risk Assessment, Detection, and Response," in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, New York, NY, USA, 2011, pp. 355–366.

[5] A. Antonini, A. Barenghi, G. Pelosi, and S. Zonouz, "Security challenges in building automation and SCADA," in 2014 International Carnahan Conference on Security Technology (ICCST), 2014, pp. 1–6.

[6] D. L. Cooper, "Data Security: Data Breaches," in Proceedings of the 2015 Information Security Curriculum Development Conference, New York, NY, USA, 2015, pp. 13:1–13:3.

[7] K. N. bin Baharin, N. M. Din, M. Z. Jamaludin, and N. M. Tahir, "Third party security audit procedure for network environment," in 4th National Conference on Telecommunication Technology, 2003. NCTT 2003 Proceedings, 2003, pp. 26–30.

[8] E. C. Lo and M. Marchand, "Security audit: a case study [information systems]," in Canadian Conference on Electrical and Computer Engineering, 2004, 2004, vol. 1, pp. 193–196 Vol.1.

[9] L. Liu, W. Jiang, and Q. Huang, "A framework for business-oriented security audit," in 2008 6th IEEE International Conference on Industrial Informatics, 2008, pp. 141–146.

[10] J. Liu, X. Wang, D. Jiao, and C. Wang, "Research and design of security audit system for compliance," in 2012 International Symposium on Information Technology in Medicine and Education (ITME), 2012, vol. 2, pp. 905–909.

[11] M. Kanatov, L. Atymtayeva, and B. Yagaliyeva, "Expert systems for information security management and audit. Implementation phase issues," in 15th International Symposium on Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 2014, pp. 896–900.

[12] Z. Han, X. Li, and E. Stroulia, "A Hierarchical Security-Auditing Methodology for Cloud Computing," in 2015 IEEE International Conference on Services Computing (SCC), 2015, pp. 202–209.

[13] W. Granzer, F. Praus, and W. Kastner, "Security in Building Automation Systems," IEEE Trans. Ind. Electron., vol. 57, no. 11, pp. 3622–3630, Nov. 2010.

[14] A. Antonini, A. Barenghi, and G. Pelosi, "Security Analysis of Building Automation Networks," in Secure IT Systems, H. R. Nielson and D. Gollmann, Eds. Springer Berlin Heidelberg, 2013, pp. 199–214.

[15] T. Mundt and P. Wickboldt, "Security in building automation systems - a first analysis," in 2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security), 2016, pp. 1–8.

[16] E. Finch, "Is IP everywhere the way ahead for building automation?," Facilities, vol. 19, no. 11/12, pp. 396–403, Nov. 2001.

[17] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A Flow-based Model for Internet Backbone Traffic," in Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment, New York, NY, USA, 2002, pp. 35–47.

[18] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural Analysis of Network Traffic Flows," in Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, New York, NY, USA, 2004, pp. 61–72.

[19] X. (George) Meng, S. H. Y. Wong, Y. Yuan, and S. Lu, "Characterizing Flows in Large Wireless Data Networks," in Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 2004, pp. 174–186.

[20] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-Based Intrusion Detection," IEEE Commun. Surv. Tutor., vol. 12, no. 3, pp. 343–356, Third 2010.

[21] R. Krejčí, P. Čeleda, and J. Dobrovolný, "Traffic Measurement and Analysis of Building Automation and Control Networks," in Dependable Networks and Services, R. Sadre, J. Novotný, P. Čeleda, M. Waldburger, and B. Stiller, Eds. Springer Berlin Heidelberg, 2012, pp. 62–73.

[22] N. Joukov, V. Shorokhov, and D. Tantsuyev, "Security audit of data flows across enterprise systems and networks," in Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for, 2014, pp. 240–247.

[23] "Index of /pub/linux/kernel/v2.6." [Online]. Available: https://www.kernel.org/pub/linux/kernel/v2.6/. [Accessed: 03-Jul-2016].

[24] "utelnetd-0.1.11-changelog." [Online]. Available: http://public.pengutronix.de/software/utelnetd/utelnetd-0.1.11-changelog. [Accessed: 13-Mar-2017].

[25] "Boa Webserver." [Online]. Available: http://www.boa.org/news.html. [Accessed: 13-Mar-2017].

[26] "Linux Linux Kernel version 2.6.32 : Security vulnerabilities." [Online]. Available: https://www.cvedetails.com/vulnerability-list/vendor_id-33/product_id-47/version_id-123682/Linux-Linux-Kernel-2.6.32.html. [Accessed: 21-Jul-2016].

[27] "CVE-2011-0709 : The br_mdb_ip_get function in net/bridge/br_multicast.c in the Linux kernel before 2.6.35-rc5 allows remote attackers to." [Online]. Available: https://www.cvedetails.com/cve/CVE-2011-0709/. [Accessed: 21-Jul-2016].

[28] "CVE - CVE-2009-4496." [Online]. Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-4496. [Accessed: 11-Jul-2016].

[29] "Pentesters (and Attackers) Love Internet Connected Security Cameras!," SANS Internet Storm Center. [Online]. Available: https://isc.sans.edu/forums/diary/Pentesters+and+Attackers+Love+Internet+Connected+Security+Cameras/21231/. [Accessed: 11-Jul-2016].

[30] "Windows XP End of Support." [Online]. Available: https://www.microsoft.com/en-us/WindowsForBusiness/end-of-xp-support. [Accessed: 21-Jul-2016].

[31] "Microsoft Windows Xp : CVE security vulnerabilities, versions and detailed reports." [Online]. Available: https://www.cvedetails.com/product/739/Microsoft-Windows-Xp.html?vendor_id=26. [Accessed: 13-Mar-2017].

[32] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," Computer, vol. 35, no. 10, pp. 54–62, Oct. 2002.

[33] S. Bhatia, N. Kush, C. Djamaludin, J. Akande, and E. Foo, "Practical modbus flooding attack and detection," in Proceedings of the Twelfth Australasian Information Security Conference-Volume 149, 2014, pp. 57–65.

[34] R. R. R. Barbosa and A. Pras, "Intrusion Detection in SCADA Networks," in Mechanisms for Autonomous Management of Networks and Services, B. Stiller and F. D. Turck, Eds. Springer Berlin Heidelberg, 2010, pp. 163–166.

[35] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," IEEE Trans. Autom. Control, vol. 58, no. 11, pp. 2715–2729, 2013.