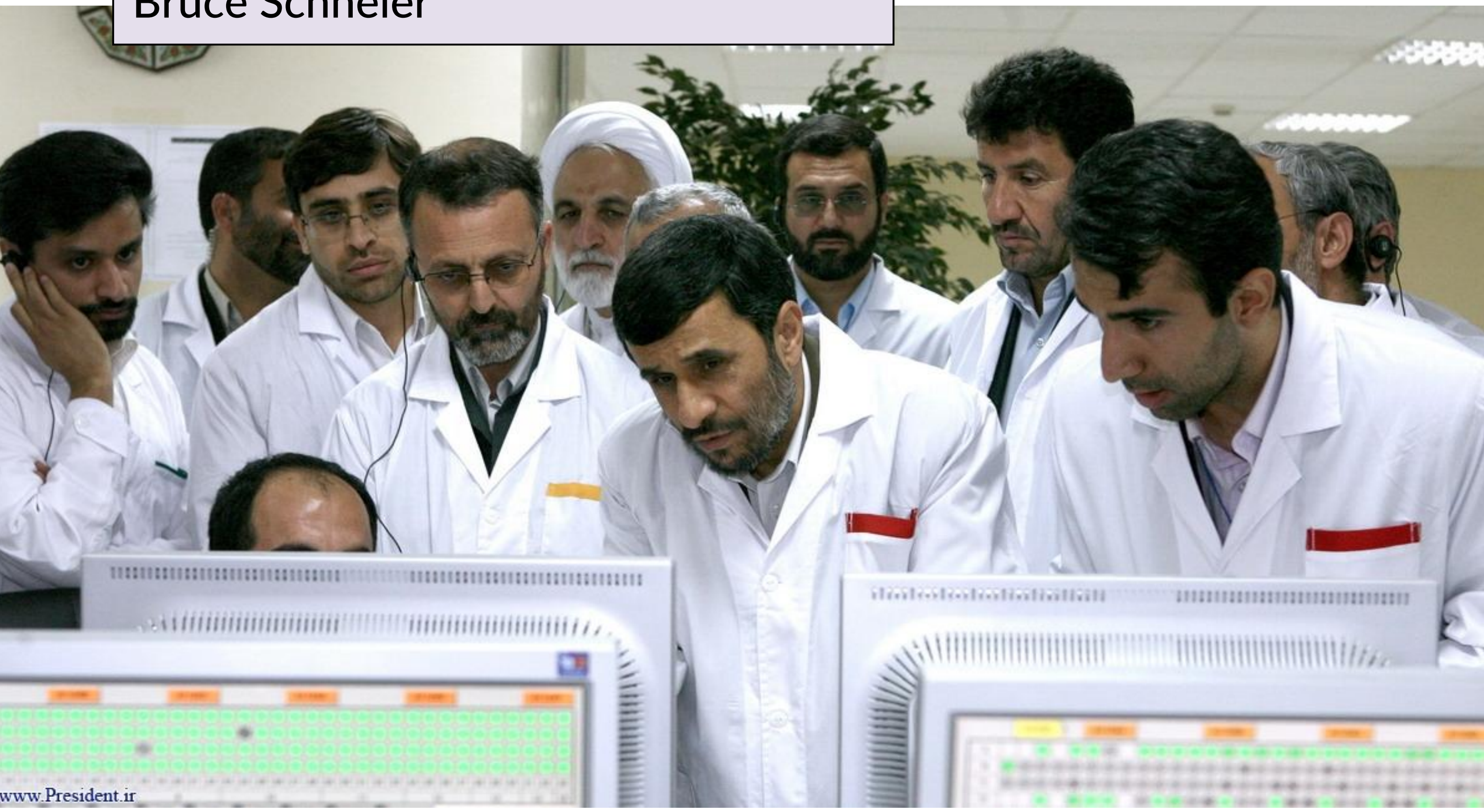
"The definitive analysis of Stuxnet"
Bruce Schneier



To Kill a Centrifuge

A Technical Analysis of What Stuxnet's Creators Tried to Achieve

Ralph Langner

November 2013

The Langner Group
Arlington | Hamburg | Munich

Content

Acknowledgements

Andreas Timm, Olli Heinonen, Richard Danzig, and R. Scott Kemp provided valuable feedback in the process of writing this paper. Nevertheless any views expressed are the author's, not theirs.

Executive Summary

This document summarizes the most comprehensive research on the Stuxnet malware so far: It combines results from reverse engineering the attack code with intelligence on the design of the attacked plant and background information on the attacked uranium enrichment process. It looks at the attack vectors of the two different payloads contained in the malware and especially provides an analysis of the bigger and much more complex payload that was designed to damage centrifuge rotors by overpressure. With both attack vectors viewed in context, conclusions are drawn about the reasoning behind a radical change of tactics between the complex earlier attack and the comparatively simple later attack that tried to manipulate centrifuge rotor speeds. It is reasoned that between 2008 and 2009 the creators of Stuxnet realized that they were on to something much bigger than to delay the Iranian nuclear program: History's first field experiment in cyber-physical weapon technology. This may explain why in the course of the campaign against Natanz, OPSEC was lossened to the extent that one can speculate that the attackers really were no longer ultimately concerned about being detected or not but rather pushing the envelope.

Another section of this paper is dedicated to the discussion of several popular misconceptions about Stuxnet, most importantly how difficult it would be to use Stuxnet as a blueprint for cyber-physical attacks against critical infrastructure of the United States and their allies. It is pointed out that offensive cyber forces around the world will certainly learn from history's first true cyber weapon, and it is further explained why nation state resources are not required to launch cyber-physical attacks. It is also explained why conventional infosec wisdom and deterrence does not sufficiently protect against Stuxnet-inspired copycat attacks.

The last section of the paper provides a wealth of plant floor footage that allows for a better understanding of the attack, and it also closes a gap in the research literature on the Iranian nuclear program that so far focused on individual centrifuges rather than on higher-level assemblies such as cascades and cascade units. In addition, intelligence is provided on the instrumentation and control that is a crucial point in understanding Iran's approach to uranium enrichment.

There is only one reason why we publish this analysis: To help asset owners and governments protect against sophisticated cyber-physical attacks as they will almost definitely occur in the wake of Stuxnet. Public discussion of the subject and corporate strategies on how to deal with it clearly indicate widespread misunderstanding of the attack and its details, not to mention a misunderstanding of how to secure industrial control systems in general. For example, post-Stuxnet mitigation strategies like emphasizing the use of air gaps, anti-virus, and security patches are all indications of a failure to understand how the attack actually worked. By publishing this paper we hope to change this unsatisfactory situation and stimulate a broad discussion on proper mitigation strategies that don't miss the mark.

Prologue: A Textbook Example of Cyber Warfare

Even three years after being discovered, Stuxnet continues to baffle military strategists, computer security experts, political decision makers, and the general public. The malware marks a clear turning point in the history of cyber security and in military history as well. Its impact for the future will most likely be substantial, therefore we should do our best to understand it properly. The actual outcome at Ground Zero is unclear, if only for the fact that no information is available on how many controllers were actually infected with Stuxnet. Theoretically, any problems at Natanz that showed in 2009 IAEA reports could have had a completely different cause other than Stuxnet. Nevertheless forensic analysis can tell us what the attackers *intended* to achieve, and how.

But that cannot be accomplished by just understanding computer code and zero-day vulnerabilities. Being a cyber-physical attack, one has to understand the physical part as well – the design features of the plant that was attacked, and of the process parameters of this plant. Different from cyber attacks as we see them every day, a cyber-physical attack involves three layers and their specific vulnerabilities: The IT layer which is used to spread the malware, the control system layer which is used to manipulate (but not disrupt) process control, and finally the physical layer where the actual damage is created. In the case of the cyber attack against Natanz, the vulnerability on the physical layer was the fragility of the fast-spinning centrifuge rotors that was exploited by manipulations of process pressure and rotor speed. The Stuxnet malware makes for a textbook example how interaction of these layers can be leveraged to create physical destruction by a cyber attack. Visible through the various cyber-physical exploits is the silhouette of a methodology for *attack engineering* that can be taught in school and can ultimately be implemented in algorithms.

While offensive forces will already have started to understand and work with this methodology, defensive forces did not – lulling themselves in the theory that Stuxnet was so specifically crafted to hit just one singular target that is so different from common critical infrastructure installations. Such thinking displays deficient capability for abstraction. While the *attack* was highly specific, attack *tactics and technology* are not; they are generic and can be used against other targets as well. Assuming that these tactics would not be utilized by follow-up attackers is as naïve as assuming that history's first DDoS attack, first botnet, or first self-modifying attack code would remain singular events, tied to their respective original use case. At this time, roughly 30 nations employ offensive cyber programs, including North Korea, Iran, Syria, and Tunisia. It should be taken for granted that every serious cyber warrior will copy techniques and tactics used in history's first true cyber weapon. It should therefore be a priority for defenders to understand those techniques and tactics equally well, if not better.

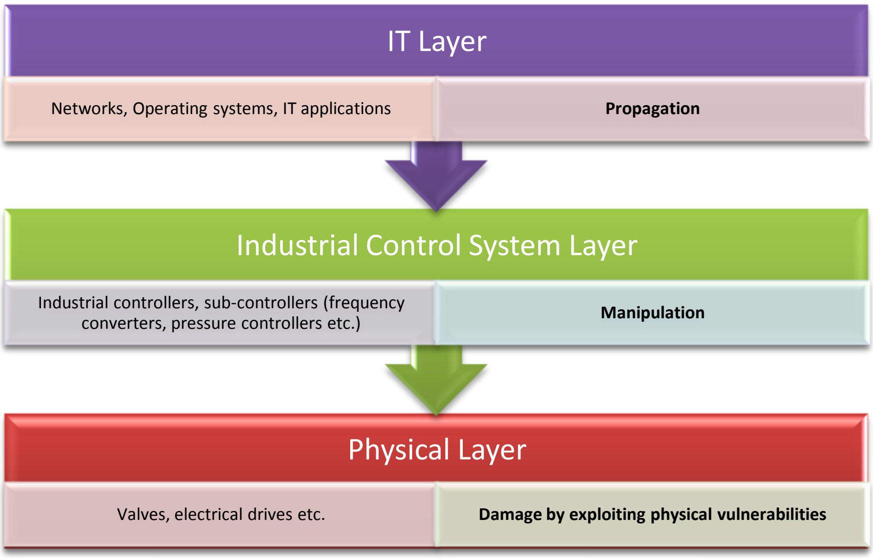

IT Layer
Networks, Operating systems, IT applications
Propagation
Industrial Control System Layer
Industrial controllers, sub-controllers (frequency converters, pressure controllers etc.)
Manipulation
Physical Layer
Valves, electrical drives etc.
Damage by exploiting physical vulnerabilities

Figure 1: The three layers of a sophisticated cyber-physical attack

A. Exploring the Attack Vector

Unrecognized by most who have written on Stuxnet, the malware contains two strikingly different attack routines. While literature on the subject has focused almost exclusively on the smaller and simpler attack routine that changes the speeds of centrifuge rotors, the "forgotten" routine is about an order of magnitude more complex and qualifies as a plain nightmare for those who understand industrial control system security. Viewing both attacks in context is a prerequisite for understanding the operation and the likely reasoning behind the scenes.

Both attacks aim at damaging centrifuge rotors, but use different tactics. The first (and more complex) attack attempts to over-pressurize centrifuges, the second attack tries to over-speed centrifuge rotors and to take them through their critical (resonance) speeds.

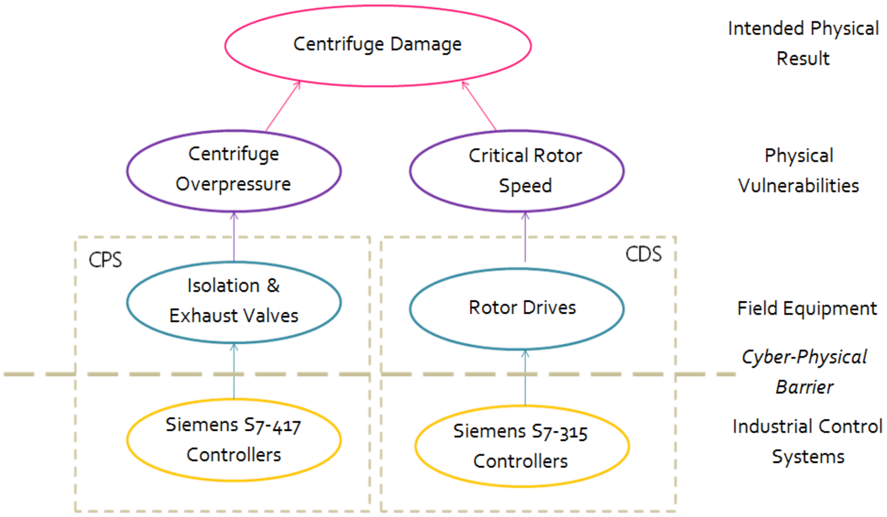


Figure 2: Synopsis of the two different attacks implemented in Stuxnet. Both use a manipulation of industrial control systems to achieve physical damage, exploiting different physical vulnerabilities of the equipment (centrifuge rotors) that basically lead to the same physical result

Overpressure Attack: Silent Hijack of the Crown Jewels

In 2007, an unidentified person submitted a sample of code to the collaborative anti-virus platform *Virustotal* that much later turned out as the first variant of Stuxnet that we know of. Whilst not understood by any anti-virus company at the time, that code contained a payload for severely interfering with the Cascade Protection System (CPS) at the Natanz Fuel Enrichment Plant.

Iran's low-tech approach to uranium enrichment

The backbone of Iran's uranium enrichment effort is the IR-1 centrifuge which goes back to a European design of the late Sixties / early Seventies that was stolen by Pakistani nuclear trafficker A. Q. Khan. It is an obsolete design that Iran never managed to operate reliably. Reliability problems may well have started as early as 1987, when Iran began experimenting with a set of decommissioned P-1 centrifuges acquired from the Khan network. Problems with getting the centrifuge rotors to spin flawlessly will also likely have resulted in the poor efficiency that can be observed when analyzing IAEA

What's a centrifuge cascade?

Gas centrifuges used for uranium enrichment are assembled into groups to maximize efficiency. Centrifuges within one group, also called an enrichment stage, share the same feed, product, and tails piping. The collective tails is then piped into the collective feed of the next stage on one side, as well as the collective product is piped into the collective feed on the other side. At the very far ends of the cascade, product and tails take-offs collect the enriched and depleted uranium. A central common feed of UF6 is entered at the "feed stage". Until 2012, Iran used cascades with 164 centrifuges grouped into 15 stages, with stage 10 as the feed stage.

reports, suggesting that the IR-1 performs only half as well – best case – as it could theoretically. A likely reason for such poor performance is that Iran reduced the operating pressure of the centrifuges in order to lower rotor wall pressure. But less pressure means less throughput – and thus less efficiency.

As unreliable and inefficient as the IR-1 is, it offered a significant benefit: Iran managed to produce the antiquated design at industrial scale. It must have seemed striking to compensate reliability and efficiency by volume, accepting a constant breakup of centrifuges during operation because they could be manufactured faster than they crashed. Supply was not a problem. But how does one use thousands of fragile centrifuges in in a sensitive industrial process that doesn't tolerate even minor equipment hiccups? In order to achieve that, Iran uses a Cascade Protection System which is quite unique as it is designed to cope with ongoing centrifuge trouble by implementing a crude version of fault tolerance. The protection system is a critical system component for Iran's nuclear program as without it, Iran would not be capable of sustained uranium enrichment.

What's a protection system?

Industrial control systems and their associated instrumentation are basically grouped into production systems and protection systems. As the name implies, protection systems don't serve any purpose during normal operation but are intended to detect process abnormalities and prevent them from destroying equipment or turning into hazards for operators and the environment.

The inherent problem in the Cascade Protection System and its workaround

The Cascade Protection System consists of two layers, the lower layer being at the centrifuge level. Three fast-acting shut-off valves are installed for every centrifuge at the connectors of centrifuge piping and enrichment stage piping. By closing the valves, centrifuges that run into trouble – indicated by vibration – can be isolated from the stage piping. Isolated centrifuges are then run down and can be replaced by maintenance engineers while the process keeps running.

The central monitoring screen of the Cascade Protection System, which is discussed in detail in the last section of this paper, shows the status of each centrifuge within a cascade – running or isolated – as either a green or a grey dot. Grey dots are nothing special on the control system displays at Natanz and even appear in the official press photos shot during former president Ahmadinejad's visit to Natanz in 2008. It must have appeared normal to see grey dots, as Iran was used to rotor trouble since day one. While no Western plant manager would have cleared such photographic material for publication, Iran didn't seem to bother to hide that fact from the media. To the contrary, there might have been a sense of pride involved by showing a technological achievement that allowed for tolerating centrifuge failure.

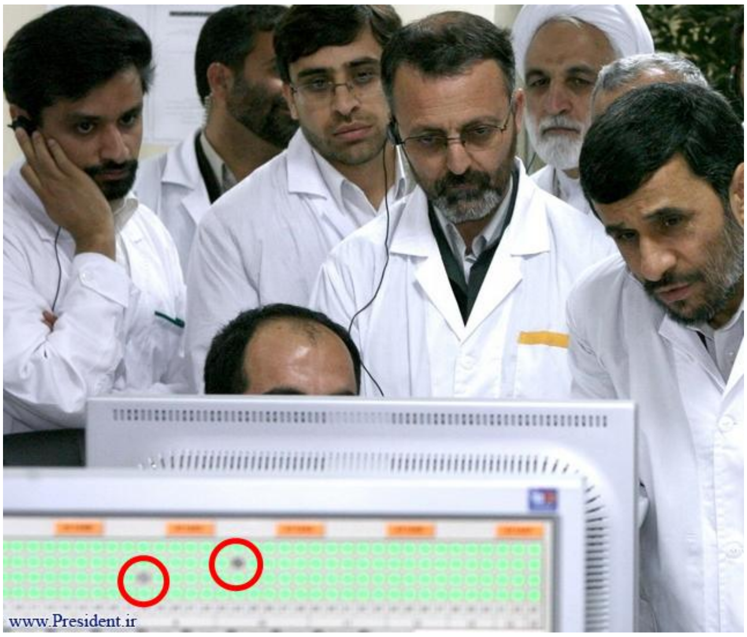


Figure 3: Former president Ahmadinejad looking at SCADA screens in the control room at Natanz in 2008. The screen facing the photographer shows that two centrifuges are isolated, indicating a defect, but that doesn't prevent the respective cascade from continuing operation (highlighting in red not in the original)

But the isolation valves can turn into as much of a problem as a solution.

When operating basically unreliable centrifuges, one will see shut-offs frequently, and maintenance may not have a chance to replace damaged centrifuges before the next one in the same enrichment stage gets isolated. Once that multiple centrifuges are shut off within the same stage, UF6 gas pressure – the most sensitive parameter in uranium enrichment using centrifuges – will increase, which can and will lead to all kinds of problems.

Iran found a creative solution for this problem – basically another workaround on top of the first workaround. For every enrichment stage, an exhaust valve is installed that allows for compensation of overpressure. By opening the valve, overpressure is relieved into the dump system. A dump system is present in any gas centrifuge cascade used for uranium enrichment but never used in production mode; it simply acts as a backup in case of cascade trips when the centrifuges must be evacuated and the "normal" procedure to simply use the tails take-off is unavailable for whatever reason. Iran discovered they can use (or abuse) the dump system to compensate stage overpressure. For every enrichment stage, pressure (controlling variable) is monitored by a pressure sensor. If that pressure exceeds a certain setpoint, the stage exhaust valve (controlled variable) is opened, and overpressure is released into the dump system until normal operating pressure is re-established – basic downstream control as known from other applications of vacuum technology.

The problem with process pressure in gas centrifuges

Gas centrifuges for uranium enrichment are extremely sensitive to increases of process pressure above near vacuum. A slight increase in pressure may affect enrichment efficiency because the pressure profile of the cascade is disturbed, lowering product flow. A moderate increase in pressure will result in more uranium hexafluoride getting into the centrifuge, putting higher mechanical stress on the rotor. Rotor wall pressure is a function of velocity (rotor speed) and operating pressure. Ultimately, pressure may cause the UF6 to solidify. At room temperature, which is the ambient condition in Natanz' cascade hall, this takes place at about 100 millibar.

The downstream control architecture with an exhaust valve per stage was most likely not acquired from the Khan network as Pakistan may not have needed it; apparently they never experienced a similar amount of unreliability. The control system technology used at Natanz did not exist back in the Eighties when Pakistan had its biggest success in uranium enrichment. The specification for the PROFIBUS fieldbus, a realtime micro-network for attaching field devices to controllers, was first published in 1993, and the controllers used for the CPS (Siemens S7-417) were introduced to the market not earlier than 1999. However, there is no evidence of a close relation between Iran and the Khan network after 1994. Lead time for the adoption of new technology such as PROFIBUS in the automation space with its extremely long lifecycles is around ten years as asset owners are reluctant to invest in new technology until it is regarded "proven" industry standard, making it unlikely that anybody would have used the new fieldbus technology for production use in critical facilities before the early years of the new millennium, just when the Khan network was shut down. But in 1998 Pakistan had already successfully tested their first nuclear weapon, obviously without the help of the new fieldbus and control technology from the German industry giant.

What's a fieldbus?

A fieldbus is a realtime micro-network for connecting automation peripherals (such as instruments, motors, or valves) to a controller. The number of stations that can be attached to a fieldbus is quite limited and often below 255. Most fieldbus variants feature one master controller, with all other stations acting as "slaves". PROFIBUS is a major European fieldbus standard promoted by Siemens. – In new plant designs, fieldbusses are progressively replaced by Ethernet, making cyber attacks against field equipment even easier.

What we do know is that when Iran got serious about equipping the Natanz site in the early years of the new millennium, they ran into technical trouble. In October 2003, the EU3 (Britain, Germany, and France) requested that Iran suspend their enrichment activities "for a period of time" as a confidence-building measure. Iranian chief negotiator Hassan Rowhani, now president of Iran, told the EU3 that Iran agreed to a suspension "for as long as we deem necessary". Two years later, Rowhani clarified that the suspension had only been accepted in areas where Iran did not experience technical problems. In 2006, Iran didn't deem the hiatus no longer "necessary" for the simple reason that they had overcome their technical trouble. This became evident when the IAEA seals at the cascades were broken and production resumed. It can be speculated that the fine-tuned pressure control that the stage exhaust valves provide was designed between 2003 and 2006.

Figure 4: The EU3 meeting in 2003 with Hassan Rowhani and the foreign ministers of Germany, France, and Britain

The SCADA software (supervisory control and data acquisition, basically an IT application for process monitoring by operators) for the CPS also appears to be a genuine development for the Natanz Fuel Enrichment Plant. To put it quite frankly, its appearance is quite amateurish and doesn't indicate signs of the many man-years of Pakistani experience. Anything "standard" that would indicate software maturity and an experienced software development team is missing. It appears like work in progress of software developers with little background in SCADA. With Iran understanding the importance of the control system for the protection system, a reasonable strategy would have been to keep development and product support in trusted domestic hands.

Messing up Iran's technology marvel

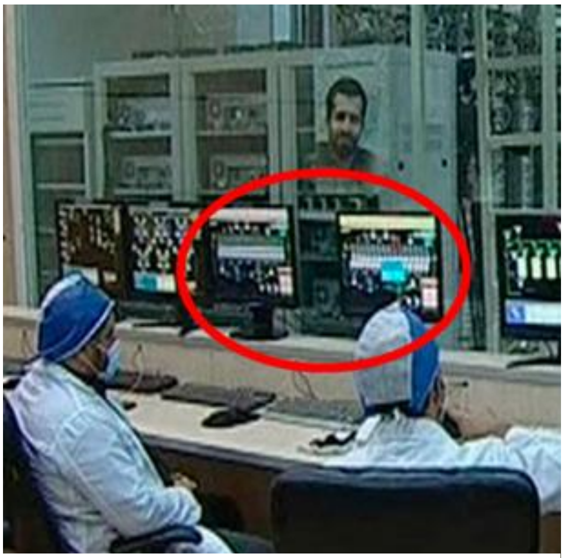


Figure 5: Operators in front of the SCADA displays of the Cascade Protection System, placed right below a picture of former president Ahmadinejad

The cyber attack against the Cascade Protection System infects Siemens S7-417 controllers with a matching configuration. The S7-417 is a top-of-the-line industrial controller for big automation tasks. In Natanz, it is used to control the valves and pressure sensors of up to six cascades (or 984 centrifuges) that share common feed, product, and tails stations.

Immediately after infection the payload of this early Stuxnet variant takes over control completely. Legitimate control logic is executed only as long as malicious code permits it to do so; it gets completely de-coupled from electrical input and output signals. The attack code makes sure that when the attack is not activated, legitimate code has access to the signals; in fact it is replicating a function of the controller's operating system that would normally do this automatically but was disabled during infection.

In what is known as a *man-in-the-middle* scenario in cyber security, the input and output signals are passed from the electrical peripherals to the legitimate program logic and vice versa by attack code that has positioned itself "in the middle".

Things change after activation of the attack sequence, which is triggered by a combination of highly specific process conditions that are constantly monitored by the malicious code. Then, the much-publicized manipulation of process values inside the controller occur. Process input signals (sensor values) are recorded for a period of 21 seconds. Those 21 seconds are then replayed in a constant loop during the execution of the attack, and will ultimately show on SCADA screens in the control room, suggesting normal operation to human operators and any software-implemented alarm routines. During the attack sequence, legitimate code continues to execute but receives fake input values, and any output (actuator) manipulations of legitimate control logic no longer have any effect.

When the actual malicious process manipulations begin, all isolation valves for the first two and the last two enrichment stages are closed, thereby blocking the product and tails outflow of process gas of each affected cascade. From the remaining centrifuges, more centrifuges are isolated, except in the feed stage. The consequence is that operating pressure in the non-isolated centrifuges increases as UF6 continues to flow into the centrifuge via the feed, but cannot escape via the product and tails take-offs, causing pressure to rise continuously.

What's SCADA?

SCADA in an acronym for Supervisory Control And Data Acquisition, a category of computer programs used to display and analyze process conditions. Poorly understood by most non-technical authors on the subject, SCADA is only one component of an automated facility and does not directly interfere with actuator devices such as valves, pumps, or motors – this is achieved by industrial controllers that operate in real time and have no display and keyboard. SCADA is the front-end of an industrial process to human operators. In the case of Stuxnet, all process manipulations occurred on controllers, not on SCADA systems.

At the same time, stage exhaust valves stay closed so that overpressure cannot be released to the dump line. But that is easier said than done because of the closed-loop implementation of the valve control. The valves' input signals are not attached directly to the main Siemens S7-417 controllers but by dedicated pressure controllers that are present once per enrichment stage. The pressure controllers have a configurable setpoint (threshold) that prompts for action when exceeded, namely to signal the stage exhaust valve to open until the measured process pressure falls below that threshold again. The pressure controllers must have a data link to the Siemens S7-417 which enables the latter to manipulate the valves. With some uncertainty left we assume that the manipulation didn't use direct valve close commands but a de-calibration of the pressure sensors.

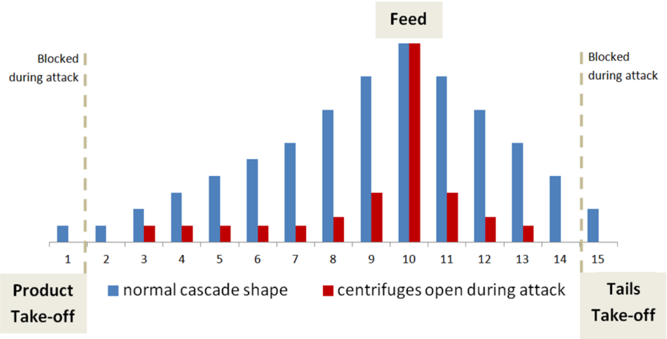


Figure 6: Modified cascade shape during the attack. Isolating all centrifuges in stage 1 and 15 effectively blocks the outflow of process gas, resulting in an increase in pressure for the non-isolated centrifuges

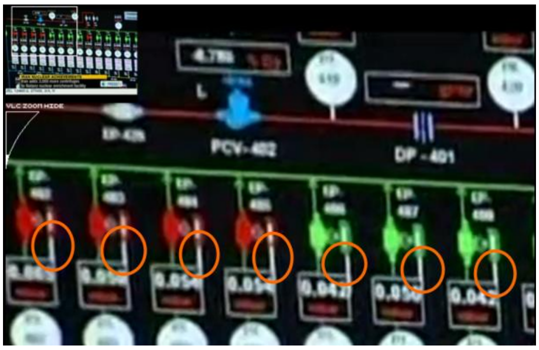


Figure 7: *Control loops* (highlighted in orange) in a SCADA display from Natanz indicate that the stage exhaust valves are controlled in a closed loop by dedicated pressure controllers

Pressure sensors are not perfect at translating pressure into an analog output signal, but their errors can be corrected by calibration. The pressure controller can be told what the "real" pressure is for given analog signals and then automatically *linearize* the measurement to what would be the "real" pressure. If the linearization is overwritten by malicious code on the S7-417 controller, analog pressure readings will be "corrected" during the attack by the pressure controller, which then interprets all analog pressure readings as perfectly normal pressure no matter how high or low their analog values are. The pressure controller then acts accordingly by never opening the stage exhaust valves. In the meantime, actual pressure keeps rising. The sensors for feed header, product take-off and tails take-off needed to be compromised as well because with the flow of process gas blocked, they would have shown critical high (feed) and low (product and tails) pressure readings, automatically closing the master feed valves and triggering an alarm. Fortunately for the attackers, the same tactic could be used for the exhaust valves and the additional pressure transducers, numbered from 16 to 21 in the facility and in the attack code, as they use the same products and logic. The detailed pin-point manipulations of these sub-controllers indicate a deep physical and functional knowledge of the target environment; whoever provided the required intelligence may as well know the favorite pizza toppings of the local head of engineering.

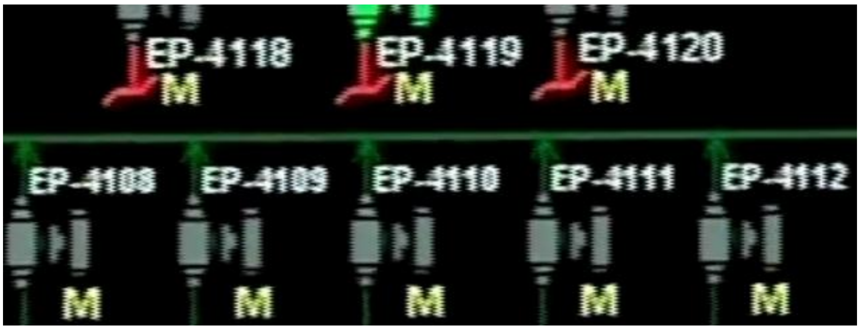


Figure 8: Very different valves: While the stage exhaust valves (labeled EP-4108 to 4112 in this partial screenshot from the CPS SCADA display) stay closed during normal operation *and during the attack*, at least one of the feed valves (labeled EP-4118 to EP-4120) must stay open. Pressure controllers at the product and tails take-offs must also be compromised to not signal a low pressure condition.

The attack continues until the attackers decide that enough is enough, based on monitoring centrifuge status, most likely vibration sensors, which suggests a mission abort before the matter hits the fan. If the idea was catastrophic destruction, one would simply have to sit and wait. But causing a solidification of process gas would have resulted in simultaneous destruction of hundreds of centrifuges per infected controller. While at first glance this may sound like a goal worthwhile achieving, it would also have blown cover since its cause would have been detected fairly easily by Iranian engineers in post mortem analysis. The implementation of the attack with its extremely close monitoring of pressures and centrifuge status suggests that the attackers instead took great care to *avoid* catastrophic damage. The intent of the overpressure attack was more likely to increase rotor stress, thereby causing rotors to break early – but not necessarily during the attack run.

Nevertheless, the attackers faced the risk that the attack might not work at all because it is so over-engineered that even the slightest oversight – or any configuration change – would have resulted in zero impact or, worst case, in a program crash that would have been detected by Iranian engineers quickly. It is obvious and documented later in this paper that over time Iran did change several important configuration details such as the number of centrifuges and enrichment stages per cascade, all of which would have rendered the overpressure attack useless; a fact that the attackers must have anticipated.

Rotor Speed Attack: Pushing the Envelope

Whatever the effect of the overpressure attack was, the attackers decided to try something different in 2009. That may have been motivated by the fact that the overpressure attack was lethal just by accident, that it didn't achieve anything, or – that somebody simply decided to check out something new and fresh.

The new variant that was not discovered until 2010 was much simpler and much less stealthy than its predecessor. It also attacked a completely different component: the Centrifuge Drive System (CDS) that controls rotor speeds. The attack routines for the overpressure attack were still contained in the payload, but no longer executed – a fact that must be viewed as deficient OPSEC. It provided us by far the best forensic evidence for identifying Stuxnet's target, and without the new, easy-to-spot variant the earlier predecessor may never have been discovered. That also means that the most aggressive cyber-physical attack tactics would still be unknown to the public – unavailable for use in copycat attacks, and unusable as a deterrent display of cyber power.

Bringing in the infosec cavalry

Stuxnet's early version had to be physically installed on a victim machine, most likely a portable engineering system, or it could have been passed on a USB stick carrying an infected configuration file for Siemens controllers. Once that the configuration file was opened by the vendor's engineering software, the respective computer was infected. But no engineering software to open the malicious file, equals no propagation.

That must have seemed to be insufficient or impractical for the new version, as it introduced a method of self-replication that allowed it to spread within trusted networks and via USB sticks even on computers that did not host the engineering software application. The extended dropper suggests that the attackers had lost the capability to transport the malware to its destination by directly infecting the systems of authorized personnel, or that the Centrifuge Drive System was installed and configured by other parties to which direct access was not possible. The self-replication would ultimately even make it possible to infiltrate and identify potential clandestine nuclear sites that the attackers didn't know about.

All of a sudden, Stuxnet became equipped with the latest and greatest MS Windows exploits and stolen digital certificates as the icing on the cake, allowing the malicious software to pose as legitimate driver software and thus not be rejected by newer versions of the Windows operating system. Obviously, organizations had joined the club that have a stash of zero-days to choose from and could pop up stolen certificates just like that. Whereas the development of the overpressure attack can be viewed as a process that could be limited to an in-group of top notch industrial control system security experts and coders who live in an exotic ecosystem quite remote from IT security, the circle seems to have gotten much wider, with a new center of gravity in Maryland. It may have involved a situation where the original crew is taken out of command by a casual *"we'll take it from here"* by people with higher pay grades. Stuxnet had arrived in big infosec.

What's an Engineering System?

Industrial controllers don't come with video screens, keyboards, and mice. Their programming is done offline on a computer system that is referred to as an "engineering system" as control system engineers don't consider themselves programmers so much but focus on the physical process functionality when configuring controllers – whatever goes wrong in programming will not result in a program crash as worst case, but in destruction of equipment.

Contemporary Engineering Systems are plain vanilla Windows PCs running a specific software application from the control system vendor. Laptops are particularly popular if only for the reason that still today many controllers are not connected to a LAN and can only be configured locally by RS-232 connection.

For sophisticated cyber-physical attacks, Engineering Systems are a prime target as they allow attack forwarding to industrial controllers.

But the use of the multiple zero-days came with a price. The new Stuxnet variant was much easier to identify as malicious software than its predecessor as it suddenly displayed very strange and very sophisticated behavior at the IT layer. In comparison, the dropper of the initial version looked pretty much like a legitimate or, worst case, pirated Step7 software project for Siemens controllers; the only strange thing was that a copyright notice and license terms were missing. Back in 2007, one would have to use extreme forensic efforts to realize what Stuxnet was all about – and one would have to specifically *look for it*, which was out of everybody's imagination at the time. The newer version, equipped with a wealth of exploits that hackers can only dream about, signaled even the least vigilant anti-virus researcher that this was something big, warranting a closer look. That happened in 2010 when a formerly not widely known Belarusian anti-virus company called VirusBlokAda practically stumbled over the malware and put it on the desk of the AV industry.

A new shot at cracking rotors

Centrifuge rotors – the major fragility in a gas centrifuge – have more than one way to run into trouble. In the later Stuxnet variant, the attackers explored a different path to tear them apart: Rotor velocity. Any attempt to overpressure centrifuges is dormant in the new version, and if on some cascades the earlier attack sequence would still execute when the rotor speed attack sequence starts, no coordination is implemented. The new attack is completely independent from the older one, and it manipulates a completely different control system component: The Centrifuge Drive System.

Figure 9: President Ahmadinejad holding a carbon fiber centrifuge rotor during his 2008 press tour at Natanz. This rotor is for the next-generation IR-2 centrifuge. Rotors used in the IR-1 that was attacked by Stuxnet are taller and built from metal.

That system is not controlled by the same S7-417 controllers, but by the much smaller S7-315. One S7-315 controller is dedicated to the 164 drives of one cascade (one drive per centrifuge). The cascade design using 164 centrifuges assembled in four lines and 43 columns had been provided by A. Q. Khan and resembles the Pakistani cascade layout. Every single centrifuge comes with its own motor at the bottom of the centrifuge, a highly stable drive that can run at speeds up to 100,000 rpm with constant torque during acceleration and deceleration. Such variable-frequency drives cannot be accessed directly by a controller but require the use of frequency converters; basically programmable power supplies that allow for the setting of specific speeds by providing the motor an AC current with a frequency as requested by the controller using digital commands. Frequency converters are attached to a total of six PROFIBUS segments for technical limitations of the fieldbus equipment (one PROFIBUS segment couldn't serve all frequency converters), all of which end at communication processors (CPs) that are attached to the S7-315 CPU's backplane. So while the attack code running on the S7-315 controller also talks to groups of six target sets (rotor control groups), there is no linkage whatsoever to the six target sets (cascades) of the overpressure attack that executes on the S7-417.

The attack code suggests that the S7-315 controllers are connected to a Siemens WinCC SCADA system for monitoring drive parameters. Most likely, an individual WinCC instance services a total of six cascades. However, on the video and photographic footage of the control rooms at Natanz no WinCC screen could be identified. This doesn't necessarily mean that the product is not used; installations might be placed elsewhere, for example on operator panels inside the cascade hall.

Keep it simple, stupid

Just like in the predecessor, the new attack operates periodically, about once per month, but the trigger condition is much simpler.

> **Troublesome centrifuge rotors**
>
> "You have to be extremely competent and expert to assemble, balance and run these machines [gas centrifuges] to full speed (63,000 rpm). I allowed it [the sale of centrifuges] as it was earlier sanctioned by Gen. Imtiaz and the Government and it would keep the Iranians happy and our friendship with them intact. That the Iranians failed to achieve any progress in 15 years, shows the complexities and extreme technical expertise required to master this technology."
>
> From A. Q. Khan's *Confession*

While in the overpressure attack various process parameters are monitored to check for conditions that might occur only once in a blue moon, the new attack is much more straightforward.

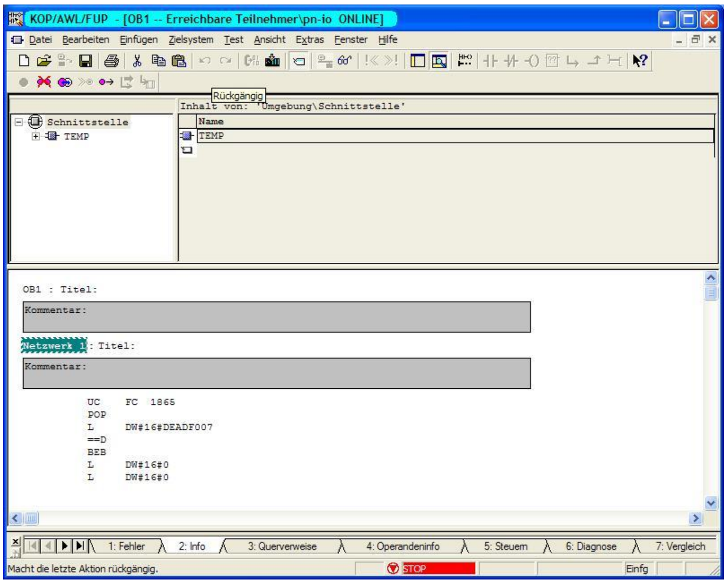
The new attack works by changing rotor speeds. With rotor wall pressure being a function of process pressure and rotor speed, the easy road to trouble is to over-speed the rotors, thereby increasing rotor wall pressure. Which is what Stuxnet did. Normal operating speed of the IR-1 centrifuge is 63,000 rpm, as disclosed by A. Q. Khan himself in his 2004 *confession*. Stuxnet increases that speed by a good one-third to 84,600 rpm for fifteen minutes, including the acceleration phase which will likely take several minutes. It is not clear if that is hard enough on the rotors to crash them in the first run, but it seems unlikely – even if just because a month later, a different attack tactic is executed, indicating that the first sequence may have left a lot of centrifuges alive, or at least more alive than dead. The next consecutive run brings all centrifuges in the cascade basically to a stop (120 rpm), only to speed them up again, taking a total of fifty minutes. A sudden stop like "hitting the brake" would predictably result in catastrophic damage, but it is unlikely that the frequency converters would permit such radical maneuver. It is more likely that when told to slow down, the frequency converter smoothly decelerates just like in an isolation / run-down event, only to resume normal speed thereafter. The effect of this procedure is not deterministic but offers a good chance of creating damage. The IR-1 is a supercritical design, meaning that operating speed is above certain critical speeds which cause the rotor to vibrate (if only briefly). Every time a rotor passes through these critical speeds, also called harmonics, it can break.

If rotors do crack during one of the attack sequences, the Cascade Protection System would kick in, isolate and run down the respective centrifuge. If multiple rotors crashed (very likely), the resulting overpressure in the stage would be compensated by the exhaust valves. Once that this would no longer be possible, for example because all centrifuges in a single stage have been isolated, a contingency dump would occur, leaving Iranian operators left with the question why all of a sudden so many centrifuges break at once. Not that they didn't have enough new ones in stock for replacement, but unexplained problems like this are any control system engineer's most frustrating experiences, usually referred to as *chasing a demon* in the machine.

Certainly another piece of evidence that catastrophic destruction was not intended is the fact that no attempts had been made to disable the Cascade Protection System during the rotor speed attack, which would have been much easier than the delicate and elaborate overpressure attack. Essentially it would only have required a very small piece of attack code from the overpressure attack that was implemented already.

OPSEC becomes less of a concern

The most common technical misconception about Stuxnet that appears in almost every publication on the malware is that the rotor speed attack would record and play back process values by means of the recording and playback of signal inputs that we uncovered back in 2010 and that is also highlighted in my TED talk. Slipping the attention of most people writing about Stuxnet, this particular and certainly most intriguing attack component is only used in the overpressure attack. The S7-315 attack against the Centrifuge Drive System simply doesn't do this, and as implemented in the CPS attack it wouldn't even work on the smaller controller for technical reasons. The rotor speed attack is much simpler. During the attack, legitimate control code is simply suspended. The attack sequence is executed, thereafter a conditional BLOCK END directive is called which tells the runtime environment to jump back to the top of the main executive that is constantly looped on the single-tasking controller, thereby re-iterating the attack and suspending all subsequent code.



KOP/AWL/FUP - [OB1 -- Erreichbare Teilnehmer\pn-io ONLINE]

```
Inhalt von: 'Umgebung\Schnittstelle'
Schnittstelle        Name
  TEMP               TEMP

OB1 : Titel:
Kommentar:

Netzwerk 1: Titel:
Kommentar:

        UC      FC  1865
        POP
        L       DW#16#DEADF007
        ==D
        BEB
        L       DW#16#0
        L       DW#16#0
```

Macht die letzte Aktion rückgängig.

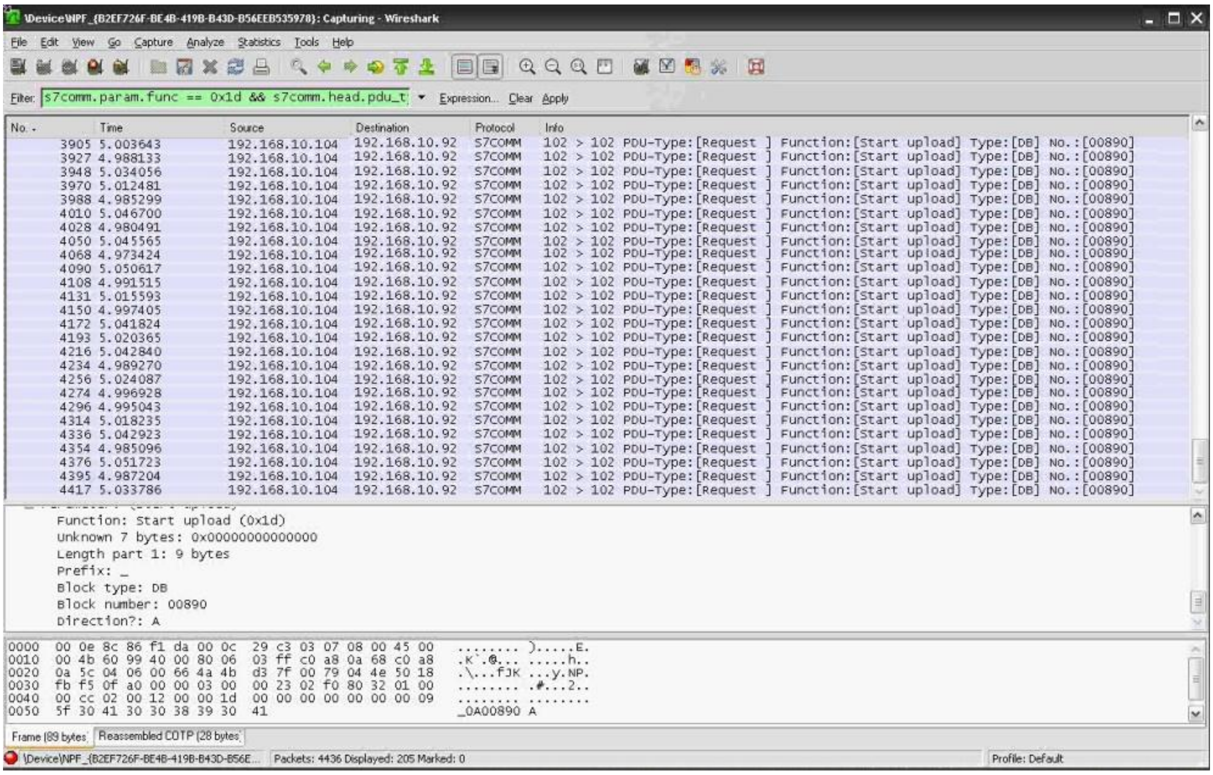
Figure 10: The attack entry point at the beginning of an infected S7-315 controller's main executive, shown in the engineering software. During attack execution, the BEB directive will disable any subsequent legitimate control logic. In comparison, the attack against the S7-417 is an order of magnitude more complex

The attackers did not care to have the legitimate code continue execution with fake input data most likely because it wasn't needed. Centrifuge rotor speed is constant during normal operation; if shown on a display, one would *expect* to see static values all the time. It is also a less dramatic variable to watch than operating pressure because rotor speed is not a controlled variable; there is no need to fine-tune speeds manually, and there is no risk that for whatever reason (short of a cyber attack) speeds would change just like stage process pressure. Rotor speed is simply set and then held constant by the frequency converter.

If a SCADA application did monitor rotor speeds by communicating with the infected S7-315 controllers, it would simply have seen the exact speed values from the time before the attack sequence executes. The SCADA software gets its information from memory in the controller, not by directly talking to the frequency converter. Such memory must be updated actively by the control logic, reading values from the converter. However if legitimate control logic is suspended, such updates no longer take place, resulting in static values that perfectly match normal operation.

Nevertheless, the implementation of the attack is quite rude; blocking control code from execution for up to an hour is something that experienced control system engineers would sooner or later detect, for example by using the engineering software's diagnostic features, or by inserting code for debugging purposes. Certainly they would have needed a clue that something was at odds with rotor speed. It is unclear if post mortem analysis provided enough hints; the fact that both overspeed and transition through critical speeds were used certainly caused disguise. However, at some point in time the attack should have been recognizable by plant floor staff just by the old ear drum. Bringing 164 centrifuges or multiples thereof from 63,000 rpm to 120 rpm and getting them up to speed again would have been noticeable – if experienced staff had been cautious enough to remove protective headsets in the cascade hall.

Another indication that OPSEC became flawed can be seen in the SCADA area. As mentioned above, it is unclear if the WinCC product is actually used to monitor the Centrifuge Drive System at Natanz. If it is, it would have been used by Stuxnet to synchronize the attack sequence between up to six cascades so that their drives would simultaneously be affected, making audible detection even easier. And if at some point in time somebody at Natanz had started to thoroughly analyze the SCADA/PLC interaction, they would have realized within hours that something was fishy, like we did back in 2010 in our lab. A Stuxnet-infected WinCC system probes controllers every five seconds for data outside the legitimate control blocks; data that was injected by Stuxnet. In a proper forensic lab setup this produces traffic that simply cannot be missed. Did Iran realize that? Maybe not, as a then-staff member of Iran CERT told me that at least the computer emergency response team did not conduct any testing on their own back in 2010 but was curiously following our revelations.


\Device\NPF_{B2EF726F-BE4B-419B-B43D-B56EEB535978}: Capturing - Wireshark
File
Edit
View
Go
Capture
Analyze
Statistics
Tools
Help
Expression...
Clear
Apply

Filter: s7comm.param.func == 0x1d && s7comm.head.pdu_t

```
No. -     Time       Source          Destination    Protocol  Info
   3905 5.003643 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   3927 4.988133 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   3948 5.034056 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   3970 5.012481 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   3988 4.985299 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4010 5.046700 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4028 4.980491 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4050 5.045565 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4068 4.973424 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4090 5.050617 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4108 4.991515 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4131 5.015593 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4150 4.997405 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4172 5.041824 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4193 5.020365 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4216 5.042840 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4234 4.989270 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4256 5.024087 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4274 4.996928 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4296 4.995043 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4314 5.018235 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4336 5.042923 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4354 4.985096 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4376 5.051723 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4395 4.987204 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
   4417 5.033786 192.168.10.104 192.168.10.92 S7COMM   102 > 102 PDU-Type:[Request ] Function:[Start upload] Type:[DB] No.:[00890]
```

```
      Function: Start upload (0x1d)
      Unknown 7 bytes: 0x00000000000000
      Length part 1: 9 bytes
      Prefix: _
      Block type: DB
      Block number: 00890
      Direction?: A
```

```
0000  00 0e 8c 86 f1 da 00 0c  29 c3 03 07 08 00 45 00   ........ ).....E.
0010  00 4b 60 99 40 00 80 06  03 ff c0 a8 0a 68 c0 a8   .K`.@... .....h..
0020  0a 5c 04 06 00 66 4a 4b  d3 7f 00 79 04 4e 50 18   .\...fJK ...y.NP.
0030  fb f5 0f a0 00 00 03 00  00 23 02 f0 80 32 01 00   ........ .#...2..
0040  00 cc 02 00 12 00 00 1d  00 00 00 00 00 00 00 09   ........ ........
0050  5f 30 41 30 30 38 39 30  41                        _0A00890 A
```

Frame (89 bytes) | Reassembled COTP (28 bytes)

\Device\NPF_{B2EF726F-BE4B-419B-B43D-B56E... Packets: 4436 Displayed: 205 Marked: 0 Profile: Default

Figure 11: Data traffic between a Stuxnet-infected WinCC SCADA system and a controller, occurring periodically every five seconds, as captured in a properly equipped forensic lab. This traffic simply could not be missed or misinterpreted by ICS security experts; it points to a cyber attack at the controller's application layer

Summing up, the differences between the two Stuxnet variants discussed here are striking. In the newer version, the attackers became less concerned about being detected. It seems a stretch to say that they *wanted* to be discovered, but they were certainly pushing the envelope and accepting the risk.

Analysis: The Dynamics of a Cyber Warfare Campaign

Everything has its roots, and the roots of Stuxnet are not in the IT domain but in nuclear counter-proliferation. Sabotaging the Iranian nuclear program had been done before by supplying Iran with manipulated mechanical and electrical equipment. Stuxnet transformed that approach from analog to digital. Not drawing from the same brain pool that threw sand in Iran's nuclear gear in the past would have been a stupid waste of resources as even the digital attacks required in-depth knowledge of the plant design and operation; knowledge that could not be obtained by simply analyzing network traffic and computer configurations at Natanz. It is not even difficult to identify potential suspects for such an operation; nuclear counter-proliferation is the responsibility of the US Department of Energy and since 1994 also of the Central Intelligence Agency, even though both organizations don't list sabotage under their official duties.

A low-yield weapon by purpose

Much has been written about the failure of Stuxnet to destroy a substantial number of centrifuges, or to significantly reduce Iran's LEU production. While that is undisputable, it doesn't appear that this was the attackers' intention. If catastrophic damage was caused by Stuxnet, that would have been by accident rather than by purpose. The attackers were in a position where they could have broken the victim's neck, but they chose continuous periodical choking instead. Stuxnet is a low-yield weapon with the overall intention to reduce the lifetime of Iran's centrifuges and make their fancy control systems appear beyond their understanding.

Reasons for such tactics are not difficult to identify. When Stuxnet was first deployed, Iran did already master the production of IR-1 centrifuges at industrial scale. It can be projected that simultaneous catastrophic destruction of *all* operating centrifuges would not have set back the Iranian nuclear program for longer than the two years setback that I have estimated for Stuxnet. During the summer of 2010 when the Stuxnet attack was in full swing, Iran operated about four thousand centrifuges, but kept another five thousand in stock, ready to be commissioned. Apparently, Iran is not in a rush to build up a sufficient stockpile of LEU that can then be turned into weapon-grade HEU but favoring a long-term strategy. A one-time destruction of their operational equipment would not have jeopardized that strategy, just like the catastrophic destruction of 4,000 centrifuges by an earthquake back in 1981 did not stop Pakistan on its way to get the bomb.

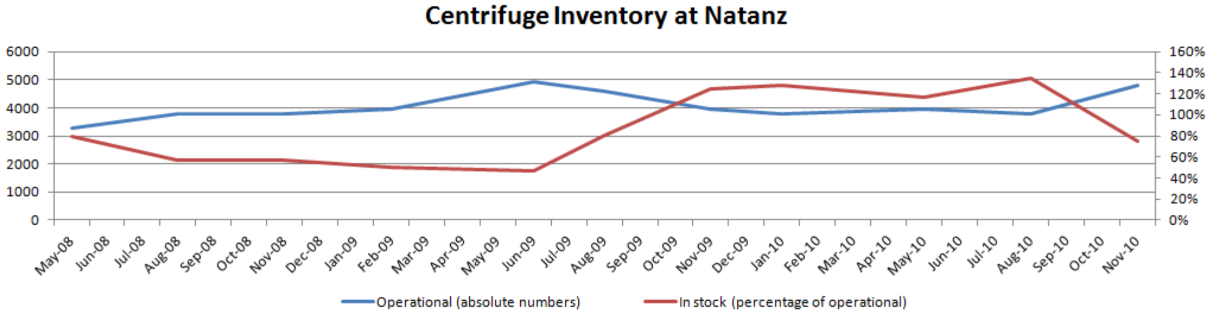


Figure 12: Centrifuge inventory at Natanz between 2008 and 2010. Iran constantly kept a stockpile of at least 50% spare centrifuges, invalidating the idea that a simultaneous catastrophic destruction of all operating centrifuges would have meant the end of the world for its nuclear ambitions

While resulting in approximately the same amount of setback for Iran as a brute-force tactic, the low-yield approach offered added value. It drove Iranian engineers crazy in the process, up to the point where they may ultimately end in total frustration about their capabilities to get a stolen plant design from the Seventies running, and to get value from their overkill digital protection system. When comparing the Pakistani and the Iranian uranium enrichment programs, one cannot fail to notice a major performance difference. Pakistan basically managed to go from zero to successful LEU production within just two years in times of a shaky economy, without the latest in digital control technology. The same effort took Iran over ten years, despite the jump-start by the Khan network and abundant money from sales of crude oil. If Iran's engineers didn't look incompetent before, they certainly did during Operation Olympic Games (Stuxnet's alleged operational code name).

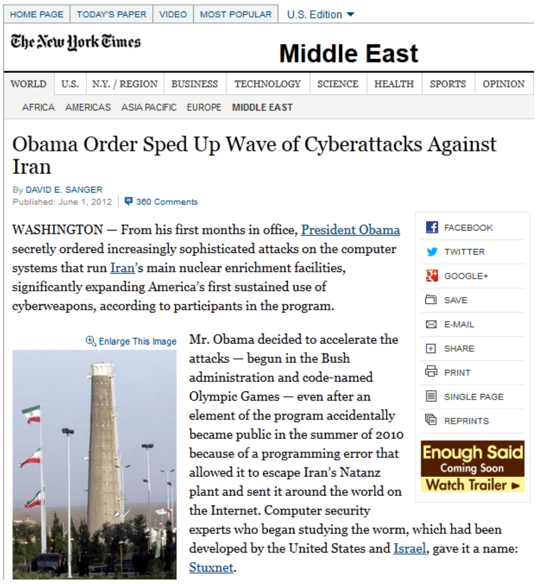
The world is bigger than Natanz

The fact that the two major versions of Stuxnet analyzed in this paper differ so dramatically suggests that during the operation, something big was going on behind the scenes. Operation Olympic Games obviously involved much more than developing and deploying a piece of malware, however sophisticated that malware may be. It was a campaign rather than an attack, and it appears like the priorities of that campaign had shifted significantly during its execution.

When we analyzed both attacks in 2010, we first assumed that they were executed simultaneously, maybe with the idea to disable the Cascade Protection System during the rotor speed attack. That turned out wrong; no coordination between the two attacks can be found in code. Then, we assumed that the attack against the Centrifuge Drive System was the simple and basic predecessor after which the big one was launched, the attack against the Cascade Protection System. The Cascade Protection System attack is a display of absolute cyber power. It appeared logical to assume a development from simple to complex. Several years later, it turned out that the opposite is the case. Why would the attackers go back to basics?

The dramatic differences between both versions point to changing priorities that will most likely have been accompanied by a change in stakeholders. Technical analysis shows that the risk of discovery no longer was the attackers' primary concern when starting to experiment with new ways to mess up operations at Natanz. The shift of attention may have been fueled by a simple insight: Nuclear proliferators come and go, but cyber warfare is here to stay. Operation Olympic Games started as an experiment with unpredictable outcome. Along the road, one result became clear: *Digital weapons work*. And different from their analog counterparts, they don't put forces in harm's way, produce less collateral damage, can be deployed stealthily, and are dirt cheap. The contents of Pandora's Box had implications much beyond Iran; they made analog warfare look low-tech, brutal, and *so Twentieth-Century*.

Somebody among the attackers may also have recognized that blowing cover would come with benefits. Uncovering Stuxnet was the end to the operation, but not necessarily the end of its utility. It would show the world what cyber weapons can do in the hands of a superpower. Unlike military hardware, one cannot display USB sticks at a military parade. The attackers may also have become concerned about another nation, worst case an adversary, would be first in demonstrating proficiency in the digital domain – a scenario nothing short of another Sputnik moment in American history. All good reasons for not having to fear detection too much.



Obama Order Sped Up Wave of Cyberattacks Against Iran

By DAVID E. SANGER
Published: June 1, 2012 | 360 Comments

WASHINGTON — From his first months in office, President Obama secretly ordered increasingly sophisticated attacks on the computer systems that run Iran's main nuclear enrichment facilities, significantly expanding America's first sustained use of cyberweapons, according to participants in the program.

Mr. Obama decided to accelerate the attacks — begun in the Bush administration and code-named Olympic Games — even after an element of the program accidentally became public in the summer of 2010 because of a programming error that allowed it to escape Iran's Natanz plant and sent it around the world on the Internet. Computer security experts who began studying the worm, which had been developed by the United States and Israel, gave it a name: Stuxnet.

Figure 13: Coming out about Stuxnet's Modus Operandi and intention: Reporting by David Sanger in the New York Times on June 1, 2012

If that twist of affairs was intentional is unknown. As with so many human endeavors, it may simply have been an unintended side effect that turned out critical. It changed global military strategy in the 21st century.

Aftermath

Whatever the hard-fact results of Stuxnet were at Ground Zero, apparently they were not viewed as disappointing failure by its creators. Otherwise it would be difficult to explain the fact that New York Times reporter David Sanger was able to find maybe five to ten high-ranking government officials who were eager to boast about the top secret operation and highlight its cleverness. It looked just a little bit too much like eagerly taking credit for it, contradicting the idea of a mission gone wrong badly.

Positive impact was seen elsewhere. Long before the coming-out but *after* Operation Olympic Games was launched, the US government started investing big time in offensive cyber warfare and the formation of US Cyber Command. The fact is that any consequences of Stuxnet can less be seen in Iran's uranium enrichment efforts than in military strategy. Stuxnet will not be remembered as a significant blow against the Iranian nuclear program. It will be remembered as the opening act of cyber warfare, especially when viewed in the context of the Duqu and Flame malware which is outside the scope of this paper. Offensive cyber warfare activities have become a higher priority for the US government than dealing with Iran's nuclear program, and maybe for a good reason. The most significant effects caused by Stuxnet cannot be seen in Natanz but in Washington DC, Arlington, and Fort Meade.

Only the future can tell how cyber weapons will impact international conflict, and maybe even crime and terrorism. That future is burdened by an irony: Stuxnet started as nuclear counter-proliferation and ended up to open the door to proliferation that is much more difficult to control: The proliferation of cyber weapon technology.

B. Misconceptions about Stuxnet's Operation and Impact

Did Stuxnet "Break Out" of Natanz due to a Programming Error?

Legend has it that in the summer of 2010, Stuxnet "escaped" from Natanz due to a software bug that came with a version update, and that the roughly 100,000 Stuxnet-infected computer systems worldwide became infected because the malware now self-propagated via the Internet much like a conventional worm. According to the story, Patient Zero was a mobile computer that a control system engineer at Natanz plugged to an infected controller, the laptop got infected and set the malware free when later connected to the Internet.

While that is a good story, it cannot be true. An infected controller contains only Stuxnet's payload and no dropper component whatsoever, making the alleged jump from controller to computer technically impossible.

All propagation routines in Stuxnet's dropper (introduced with the rotor speed attack) are carefully crafted, with the problem to be solved apparently being that physical contact to a trusted carrier had been lost. But propagation can only occur between computers that are attached to the same logical network or that exchange files via USB sticks. The propagation routines never make an attempt to spread to random targets for example by generating random IP addresses. Everything happens within the confined boundaries of a trusted network. However, these days such a trusted environment isn't necessarily local anymore. Contractors working at Natanz work for other clients as well, and they will have carried their Stuxnet-infected laptop computers to those clients and connected them to their (maybe even air-gapped) "local" networks. Patient One, let's say a cement plant, will have other contractors besides the one that employs Patient Zero, who also connect their mobile computers to the now-infected "local" network. Those will carry the malware farther. At some link in the chain, infected contractors and/or asset owners will use remote access via VPN, allowing the virus to travel over continents. All of a sudden, Stuxnet made its way around the globe, but not because of the Internet, but because trusted network connections are tunneled through the Internet these days, extending to shared folder access, however ill-advised that may be from a security perspective.

Given the fact that Stuxnet reported IP addresses and hostnames of infected systems back to its command-and-control servers, along with basic configuration data, it appears that the attackers were clearly anticipating (and accepting) a spread to non-combatant systems, and quite eager to monitor it closely – which would eventually also deliver information on contractors working at Natanz, on their other clients, and maybe even about clandestine nuclear facilities in Iran.

Did the Attackers Have the Capability to Stop the Campaign?

Speculations about the attackers' considerations to stop the campaign only to get overruled by a presidential decision to keep going miss a critical point: The attackers simply lacked the technical capability to call the attack off.

For infected engineering systems (the computers that are used to configure the industrial controllers), with or without the ability to connect to the CC servers, there is no logic implemented in the malware which could actively disable the malicious code on infected controllers. This could only have been achieved by forcing exhaustive controller re-configuration with legitimate code only, but that was out of the reach for the attackers – short of a friendly phone call to Natanz or Tehran, telling control system engineers to do just that. All one would have needed to do is make sure that the computers used for re-configuration were clean, which didn't even afford sophisticated anti-virus software but could be done simply by checking for the presence of a malicious file (s7otbxsx.dll) by a simple filename search, using nothing but software tools (Explorer) available as part of the operating system.

What the attackers could have attempted if they wanted to, was to discontinue injecting new attack routines. But all online control was lost anyway in August 2010, when Iran's national telecommunications provider blocked Internet communications to the command-and-control servers that had been used by the attackers to monitor and modify the campaign. After that date, Stuxnet was all on its own, executing autonomously. But that was what it was designed for in the first place.

Can Stuxnet be used as a Blueprint for Copycat Attacks?

Even though the tactics and exploits used by Stuxnet at the control system level are so far-out that one could speculate if its creators were on drugs, sober analysis reveals a solid systematic approach behind the implementation.

A methodology for cyber-physical attack engineering

The post-Stuxnet cyber attack engineer looks at the plant and its control systems in a holistic way, trying to identify *physical* vulnerabilities and ways to reliably exploit such vulnerabilities by cyber manipulations. Often, physical vulnerabilities are typical for a production process and plant configuration. In the case of Natanz, the physical vulnerability waiting to be exploited is the fragility of centrifuge rotors. This has been known for a long time and didn't require lots of research by Stuxnet's creators. To get a crack at physical vulnerabilities for other targets one would first look at HAZOP and similar safety analyses, and for the presence of any protection and safety systems. Different from production controllers, a protection system (while often running on identical hardware) is not needed to keep the process running. It is used to prevent process setups from damaging equipment or worse. Where such damage can include harm to humans or the plant environment, protection systems are usually referred to as safety systems that are often required by regulation, be it by OSHA, NRC or other regulators. Safety systems feature additional reliability by providing extended means to ensure code integrity and availability (such as redundancy and fault tolerance). However, all these features were never designed to withstand a cyber attack.

For Natanz, the way to exploit the physical vulnerability is to overpressure the centrifuges or to manipulate rotor speeds, resulting in predictable damage. Since centrifuge operating pressure at Natanz is controlled by the Cascade Protection System and rotor speed by the Centrifuge Drive System, these two systems became prime candidates for compromise. Only then started the cyber part of the attack engineers' work. If they are able to determine cyber manipulations which reliably exploit a physical vulnerability, they have arrived at what I call a *plant-level vulnerability*, for which Stuxnet gives the perfect example. Getting there requires looking at cyber and physical systems in the context of the plant and its physical processes; an approach waiting to be adopted in cyber defense.

Ignoring zero-days in industrial control systems

Attack engineering is about reliably taking over control in order to exploit physical vulnerabilities. The way to get there is completely different from IT. At the control system level, Stuxnet did not exploit any zero-day vulnerabilities, buffer overflows or other fancy geek stuff, but legitimate product features. In the industrial control system space, the worst vulnerabilities are not bugs, they are features. No search for buffer overflows is necessary or useful; a thorough understanding of products and their architecture is. From the attacker's point of view, exploiting flaws rather than bugs has a significant advantage: They will not be fixed over night by a vendor releasing a "patch", and users rolling out the patch quickly. Instead, the attacker can be confident that those vulnerabilities are here to stay for years, even after successful exploits are out in the wild.

To be more specific, Stuxnet teaches potential cyber attackers how to inject malicious code on realtime controllers, which may be done in the very same manner by hijacking a driver DLL or, in a more direct way, by directly talking to networked controllers without the need to compromise an engineer's workstation. It teaches how to takeover control from a legitimate program that remains running on a controller by placing malicious code at the very beginning of the main executive. It teaches how to disable legitimate control code by calling a simple jump directive. It teaches how controller library functions can be hijacked and modified. It teaches how to provide legitimate control code, and any SCADA applications as well, with fake sensor data by modifying the input process image with a simple memory write operation. It teaches how to directly interface with field equipment attached via PROFIBUS. It teaches how to disable controller cycle time monitoring by writing a simple BLOCK END directive to the respective interrupt handler. It teaches how to compromise sub-controllers by re-configuration, and how to blindfold sensors by de-calibration. That's a wealth of knowledge, put on the street by the attackers, waiting to be copied and ultimately being crafted into malware tools, making it available by point-and-click.

Indirect infiltration via soft targets

At the operational level, Stuxnet highlighted the royal road to infiltration of hard targets. Rather than trying to infiltrate directly by crawling through fifteen firewalls, three data diodes, and an intrusion detection system, the attackers played it indirectly by infecting soft targets with legitimate access to Ground Zero: Contractors. Whatever the cyber security posture of contractors may have been, it certainly was not at par with the Natanz Fuel Enrichment facility. Getting the malware on their mobile devices and USB sticks proved good enough as sooner or later they would physically carry those on site and connect them to the FEP's most critical systems, unchallenged by any guards.

Any follow-up attacker will explore this infiltration method when thinking about hitting hard targets. The sober reality is that at a global scale, pretty much every single industrial or military facility that uses industrial control systems at some scale is dependent on its network of contractors, many of which are very good at narrowly-defined engineering tasks, but lousy at cyber security. While industrial control system security had discussed the insider threat for many years, insiders who unwittingly help to deploy a cyber weapon had been completely off the radar. Obviously, they play a much more important role than the very small subset of insiders that may theoretically develop malicious intentions.

Are Nation-State Resources Required to Pull off Similar Attacks against the US or Their Allies?

It has often been stated that similar attacks against US (or other friendly) targets would require nation-state resources. From a technical perspective, this is not true. The development of Stuxnet *did* require nation-state resources – especially for intelligence gathering, infiltration, and most of all for testing. The technical analysis presented in this document clearly indicates that a) the cyber weapon was way too complex to warrant any hope for successful operation without thorough testing, and b) that such testing must have involved a fully-functional mockup IR-1 cascade operating with real uranium hexafluoride because both overpressure and rotor speed manipulations have completely different effects if executed on empty centrifuges. Obviously, a fully-functional uranium enrichment test bed that replicates a top secret plant is beyond the reach of organized crime and terrorists. But there are more copycat scenarios than the (quite silly) idea that adversaries could impact the operation of a uranium enrichment facility in the US and disguise such an attack as random equipment failure.

Trading sophistication and reliability for scale

It is quite unreasonable to expect a sophisticated cyber attack against a similar singular high-value US target, at least not in time of peace. That doesn't mean we're safe. Attack technology can and should be separated from attack scenarios with their specific objectives and constraints. Assuming that adversaries will try to maximize cost/benefit ratio, they will most likely focus on targets that are much easier to attack using lessons learned from Stuxnet – targets that are plentiful and accessible and much easier to attack, such as critical infrastructure installations. Not only is civilian critical infrastructure a more promising target for adversaries because of better accessibility, but also because of standardization. Even A. Q. Khan did not sell turnkey uranium enrichment plants which are used in hundreds of locations in different countries. For power plants, electrical substations, chemical plants and the like, that's a different story. All modern plants operate with standard industrial control system architectures and products from just a handful of vendors per industry, using similar or even identical configurations. This has implications that are much more important than the increasing network connectivity that is often identified as the biggest ICS security problem.

First, intelligence gathering isn't particularly difficult. A good control system engineer that thoroughly understands the architecture and functionality of control system X for power plant A will be able to use most of his knowledge in power plant B or C as long as they use the same product and version, as one can easily tell just by looking at recruitment ads. Knowing that control system engineers are faced with comparatively low salaries and unpleasant shift work makes them a source of relevant skills that can be drained easily; an approach that is much more promising than training hackers in industrial control systems and plant operations.

Second, once that attack tactics are identified and implemented, they can be used not just to hit one specific target, but multiple targets. A simultaneous low-key attack against multiple targets can have as much of an effect as a much more costly and sophisticated attack against a singular high-value target. Attack sophistication and reliability can be traded for scalability. It gives any potential attacker more bang for the buck if exploit code is not used exclusively against one specific target (such as an electrical substation, or water plant) but against multiple targets of the same breed, thereby achieving emergent destructive capability. As an example, a cyber attack against one power station (or electrical substation) is pretty much pointless as it has little to zero impact on grid reliability. A simultaneous attack against multiple stations can, however, result in a cascading grid failure. Adversaries beyond the script kiddie level will have figured that out already.

One of the toughest challenges is the fact that exploit code can be packaged into software tools. The genius mastermind is needed only for identifying vulnerabilities and designing exploits. Any software shop, no matter if government-driven, privately held, or in the criminal underground, would not implement such exploits as custom spaghetti code carefully adjusted to a single piece of malware, but use an object-oriented, modular approach. At some level of software maturity, such exploit components can be made available in user-friendly point-and-click software applications, just like it is now for boilerplate malware development. The skill set for those who assemble and deploy a specific sample of cyber-physical attack code will then drop dramatically.

The cost of self-imposed constraints

Other factors that made the development of Stuxnet particularly costly and should not be expected in copycat attacks were the self-imposed constraints of the attackers. Stuxnet's developers decided damage should be disguised as reliability problems. I estimate that well over 50% of Stuxnet's development cost went into efforts to hide the attack. Stuxnet-inspired attackers will not necessarily place the same emphasis on disguise; they may *want* the victim to know that they are under cyber attack, and perhaps even publicly claim credit for it. Such thinking would certainly not limit itself to the use of low-yield cyber weapons. It appears a stretch to assume that adversaries would be as concerned about collateral damage as US cyber forces, or would go so far to involve lawyers in their team for advice how to not violate international law. In the industrial control system space, an open attack doesn't even preclude follow-up attacks, as attempts to protect the targets and similar potential targets may take well over a year, allowing the attackers to strike again, maybe with fine-tuned exploits.

In order to estimate resources required for substantial Stuxnet-inspired cyber-physical attacks, one should first get credible scenarios straight. Credible scenarios involve simultaneous or staged cyber attacks against targets in critical infrastructure and manufacturing. Such targets can be hit by a Stuxnet-inspired copycat attack without requiring nation-state capabilities. The question why

America's adversaries didn't try to achieve that already is as difficult to answer as why we didn't see terrorists fly passenger airplanes into buildings before 9/11. We simply don't know. What we do know is that the capabilities of potential cyber attackers are on the rise, and at the same time vulnerabilities of potential targets for cyber-physical attacks are increasing due to a rush to more connectivity and convenience. Not a promising development.

Can Technical Security Controls Block Stuxnet-Like Attacks?

Readers familiar with cyber security and in some way associated with industrial control systems will have come across a plethora of cyber security solutions that allegedly protect critical infrastructure against Stuxnet-like attacks. In fact it has become more difficult to spot solutions that would not pretend to do the trick. Yet most of what is advertised is unsubstantiated marketing vapor.

Anti-virus software doesn't help against a Stuxnet-like attack for a simple reason. It is based on identifying and blocking known malware that is listed in the AV solution's signature database. Unfortunately there will be no signature for custom-built malware that doesn't display any strange behavior on average computer systems. As a case in point, the first Stuxnet variant was kind of rubbed into the face of the AV industry in 2007 but was identified as malware not earlier than six years later, using the knowledge gained from analyzing later variants. Malware designed like this first version is pretty much indistinguishable from a legitimate application software package and thereby flying below the radar of anti-virus technology. Even the next version with the rotor speed attack, loaded with zero-day exploits, travelled at least a year in the wild until discovered by the anti-virus industry.

Network segregation by firewalls, data diodes, air gaps and the like is a good thing per se, but not sufficient to solve the problem. In respect to recommending air gaps as a remedy, one cannot but be stunned about such ignorance of one of the most basic lessons learned from Stuxnet. Stuxnet actually demonstrated how air gaps of high-value targets can be jumped, namely by compromising mobile computers of contractors who enjoy legitimate physical access to the target environment. Since such access is often achieved locally by walking down to the respective control system cabinet, or benefits from proper authorization if performed via networks, filtering and blocking network traffic is insufficient to protect high-value targets.

The same must be said about intrusion detection and intrusion prevention systems. From a technical point of view, the intriguing idea to detect sophisticated cyber-physical attacks in network traffic is completely unvalidated. In this respect, the US Department of Defense's claim of *defending the nation at network speed* certainly does not extend to cyber-physical attacks. Defending against them cannot be done in milliseconds, it requires years of organizational and architectural changes in potential target environments.

Application of security patches doesn't necessarily do the trick either, at least when it comes to industrial control systems. While the operating system vendor was quick to deliver security patches for the zero-day vulnerabilities exploited at the OS level, the same strategy cannot be expected at the ICS application level. For example, the vendor of the ICS engineering software initially disputed any vulnerabilities in his software. Two years later, a vulnerability report was filed (CVE-2012-3015) and a patch was provided for one of the vulnerabilities that Stuxnet's dropper had exploited, namely the ability to execute arbitrary code at admin privilege by exploiting a legitimate configuration functionality of the software package. Two years may be a little bit late for exploits that don't just affect singular targets in hostile countries but thousands of targets at home. For other vulnerabilities that had been exploited by Stuxnet, such as faking sensor values by overwriting the input process image, or hijacking a driver DLL in order to inject malicious code on controllers, still no "patch" is available. In the industrial control system space, a culture to identify and correct security vulnerabilities, no matter if they are programming bugs, design flaws, or just legitimate program features introduced for convenience, waits to be adopted as best practice.

Once that the risk of cyber-physical attacks against critical infrastructure was highlighted by Stuxnet, the search for magic "silver bullets" had begun. Seemingly, the most elegant way is to solve the problem by not changing much other than applying technical point solutions. As has been pointed out in this paper, it can be demonstrated that such solutions don't do much good except for those who sell them. Stuxnet has presented cyber defense a task that cannot be mastered by simply relying on conventional infosec wisdom.

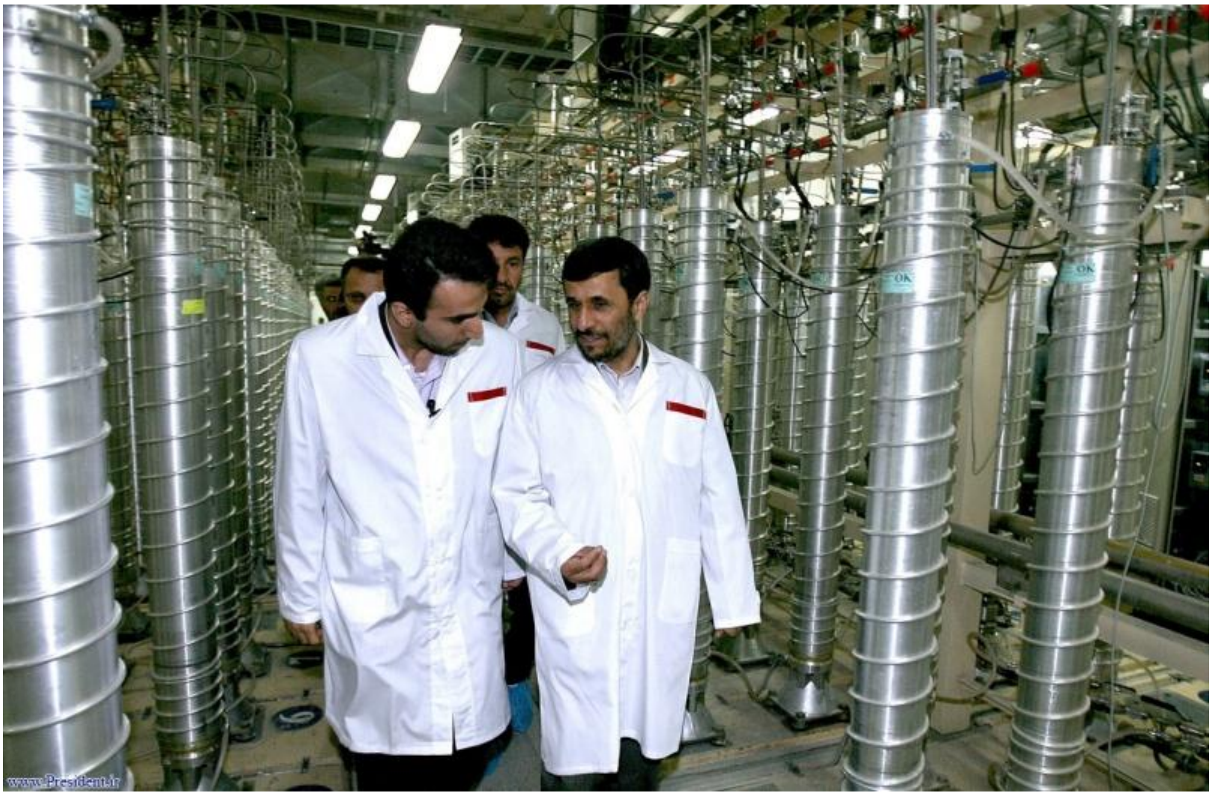
Is "Active Defense" Against Cyber-Physical Attacks Sufficient?

So far, the defensive approach of Western nations against sophisticated cyber-physical attacks in the wake of Stuxnet has been based on two assumptions. First, that such attacks would require nation-state resources; a clear misconception as has been pointed out above. Second, speculations about adversaries' motivations, and how such motivations can be anticipated or even controlled, were interpreted to suggest that substantial passive defense is not necessary.

In the tradition of risk-based thinking that factors-in threat intelligence it seemed validated to ask "who would attack us with cyber weapons, and why", and if no good answers can be found to that question, to conclude that the risk of attack must be very low. For those who believe that it still needs to be addressed, the default answer then is to attempt changing adversaries' motivation by deterrence. Unfortunately, it cannot be demonstrated that such deterrence will impress non-state actors.

The minority (including this author) believes that basing national security on theories about adversaries' motivations and wishful thinking on how to control them is a risky gamble. It advocates working towards effective passive defense "just in case", making substantial cyber-physical attacks against critical infrastructure if not impossible, much more difficult, and certainly difficult enough to put them out of reach for non-state actors. Such is a goal that is realistically achievable for those willing to accept the challenge presented by Stuxnet to start over and find and implement new and creative defensive solutions that render cyber weapons pretty much useless. Such solutions conflict with the objectives of cyber warriors not only abroad but also at home. It therefore has to be understood and addressed that these solutions will not automatically be welcomed by our own offensive cyber forces. This conflict of interest can presently not be resolved technologically but only politically. It has often been stated that cyber offense has an advantage over cyber defense. While it can be debated that this is true in technical terms in the domain of industrial control system security, it certainly does apply in a political context. Cyber offense is well-funded and implemented straightforward within a military chain of command. At the same time, cyber defense of critical national infrastructure is expected to be implemented voluntarily by a dispersed private sector that feels little desire to address matters of national security by ill-coordinated risk management exercises that negatively affect the bottom line.

C. Inside Natanz: A Guided Tour of Plant Systems, Instrumentation, and Control



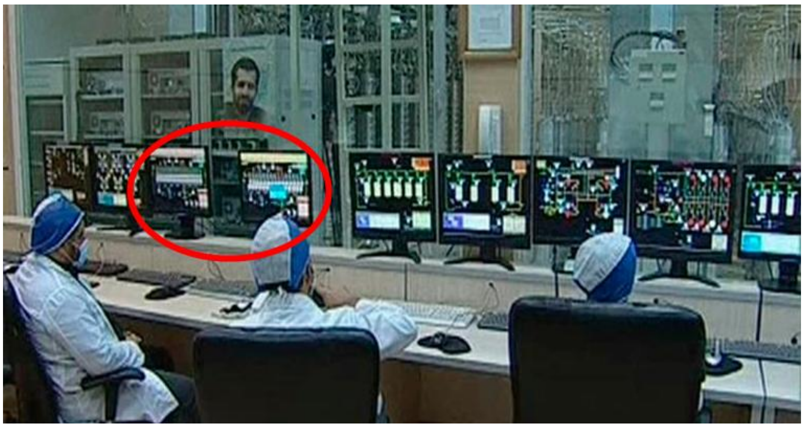
When we started our research on Stuxnet I was under the impression that design details of the Natanz Fuel Enrichment Plants were top secret and thus out of our reach. In the meantime we discovered that much to the contrary, Iran seems to be eager to release detailed footage in the open which allows analysts to arrive at a fairly good understanding of plant details, and thereby at a better understanding of Stuxnet's purpose. I also realized that while much scientific literature is available on the centrifuges, little to nothing is available on instrumentation and control. Our findings are documented here in depth in order to close this gap in research literature.

Most of the pictures presented here are taken from frame-by-frame analysis of plant floor footage that was aired on Iranian television and somehow made its way into the Internet. Others, like the picture above, are from the official media tour of President Ahmadinejad at the Natanz Fuel Enrichment Plant in 2008. As can be recognized by looking at the piping, floor markings and empty cascade stand to the right, the president is standing right at centrifuge column number four at enrichment stage four, near the product end.

SCADA Software

A wealth of intelligence can be gathered by analyzing the SCADA displays that Iran seems to show on domestic TV with a sense of pride. Essential details of the plant layout are displayed on screen. A SCADA screen is usually organized to mimic the physical and/or functional layout of the plant, such as piping, and location of important system components. Engineers refer to that as a *Piping and Instrumentation Diagram* (P&ID). Until now, this resource hasn't been tapped in research literature, maybe because the bulk of research so far had been done by nuclear scientists rather than by control system engineers.

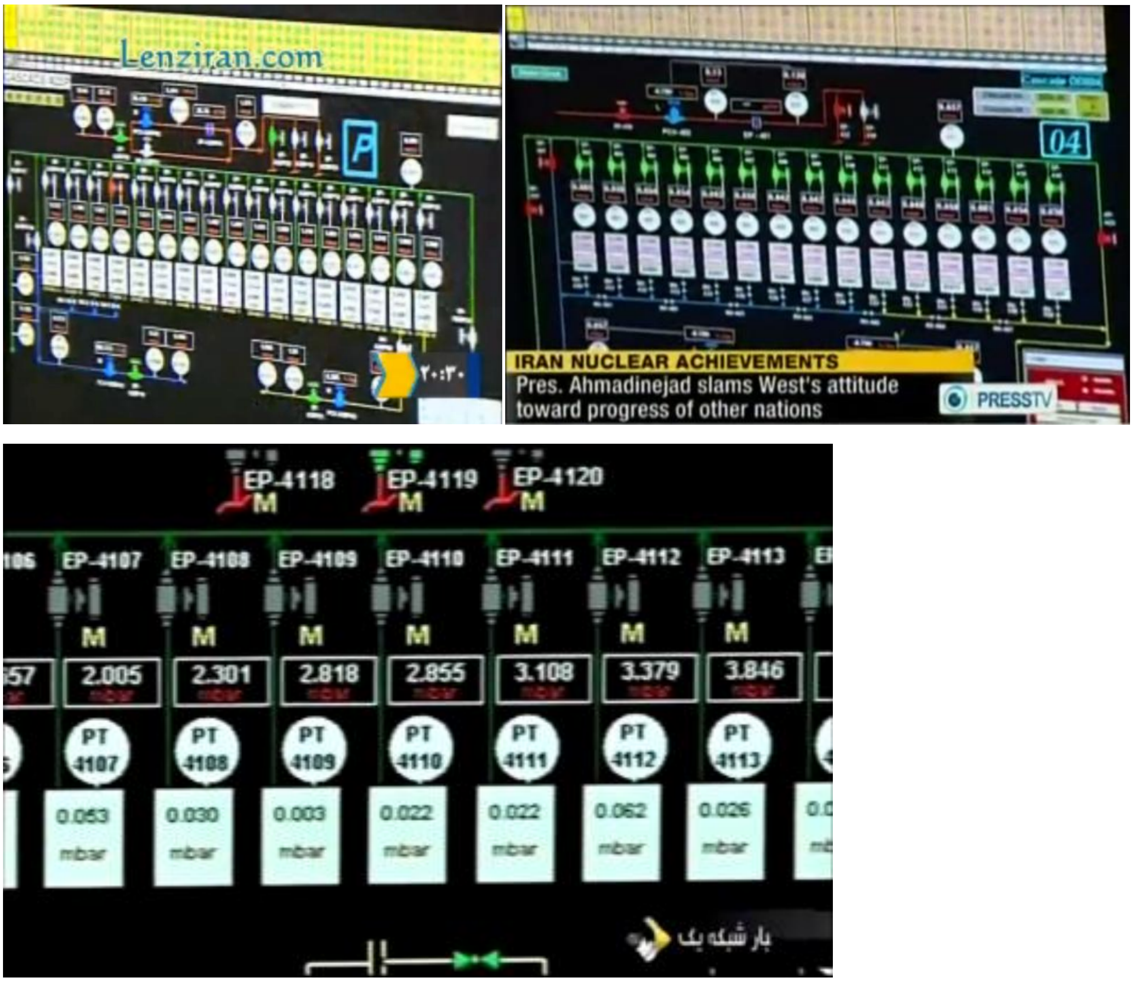
Control room



The control room of the above-ground Pilot Fuel Enrichment Plant (PFEP) as of February 2012, with operators sitting in front of SCADA screens. The two displays highlighted in red run the monitoring application for the Cascade Protection System that is discussed in-depth in this document.

The above picture shows another view of the PFEP's control room, with MIT graduate and then-president of the Iranian Atomic Energy Organization Ali Akbar Salehi at the keyboard, starting up a newly commissioned cascade. (Salehi later became vize president and foreign minister of Iran.) In the video, the scene is accompanied by heroic music and *Allahu akbar* exclamations by participants. The picture was taken in February 2010 when the Stuxnet attack was in full swing. Notes on the pink stick-on marks on the video displays are unreadable; in Western facilities, such marks would most likely identify login credentials.

The Cascade Protection System monitoring application



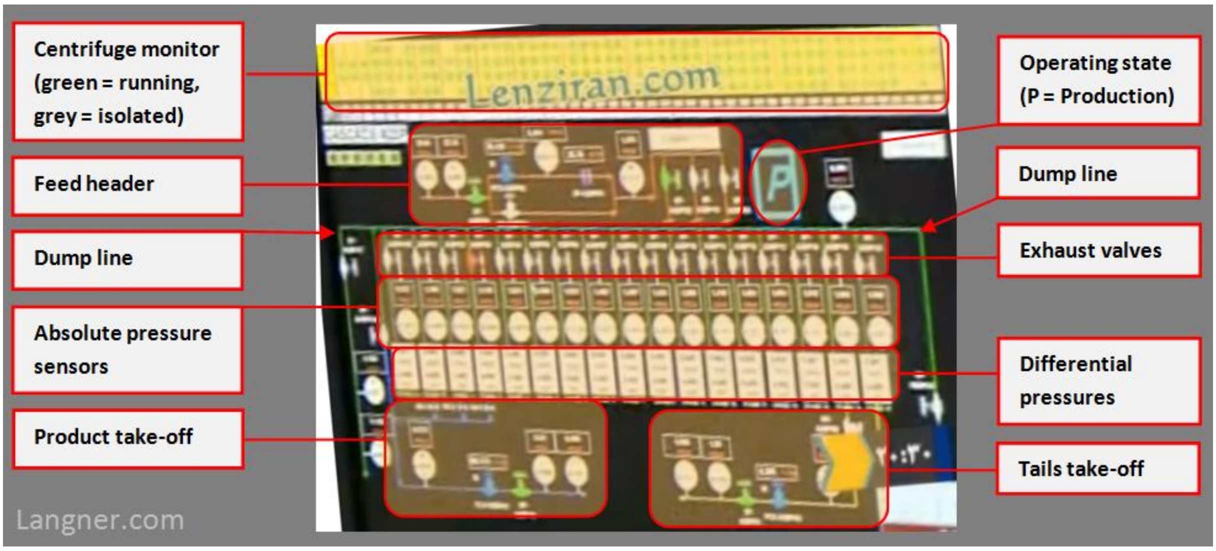
IRAN NUCLEAR ACHIEVEMENTS
Pres. Ahmadinejad slams West's attitude toward progress of other nations

The monitoring screen for the Cascade Protection System, shown above, shows the basic piping, valves, and pressure sensors of the cascades. Red piping (upper display area) signifies the feed header. Blue piping (lower left display area) signifies the product take-off, white piping (lower right display area) signifies the tails take-off, and green piping (upper display area, extending down left and right at the display borders) the pressure normalization and dump system.

Millibar readings in the rectangular black boxes (with "mbar" in red") identify absolute pressure in the respective enrichment stage. The millibar readings in the white boxes stand for differential pressure and will most likely identify the delta between actual pressure and setpoint. An operator would observe the latter to spot potentially harmful trends, which could be identified by continuous high positive readouts. In contemporary Western SCADA software, one would most likely see such information displayed graphically.

Centrifuge isolation valves are not shown, but their status can be determined in the centrifuge monitor area on top of the display. The centrifuge monitor area also allows to identify cascade shape easily. After we had discovered and published that fact in 2011, highlighting enrichment stage borders by vertical red lines in a screenshot, Iranian engineers must have thought that was a good idea and incorporated that in their software. The vertical bars in the screenshots above are not inserted by us but appear in the original footage, suggesting that Iran really doesn't care much about keeping their cascade shapes classified.

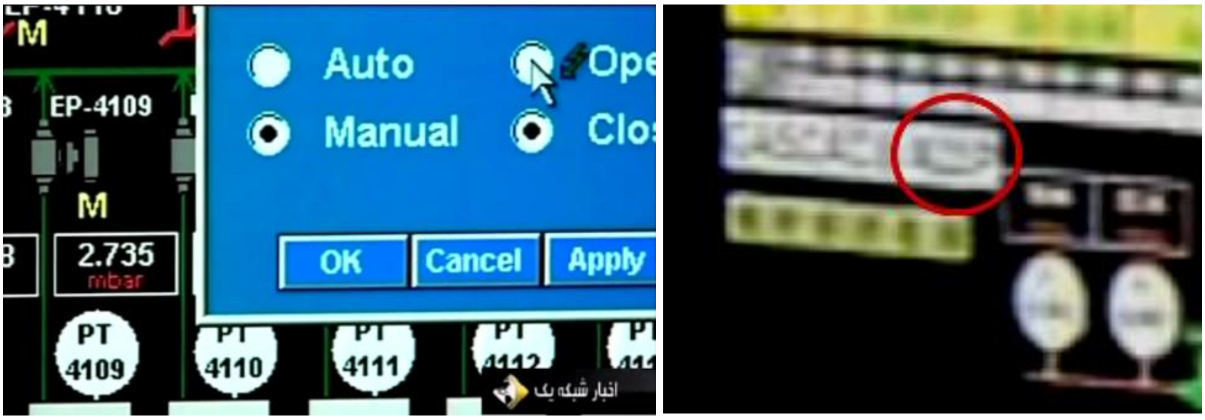
The following schematic gives an orientation of the application layout.



The software shows the status for one particular cascade.

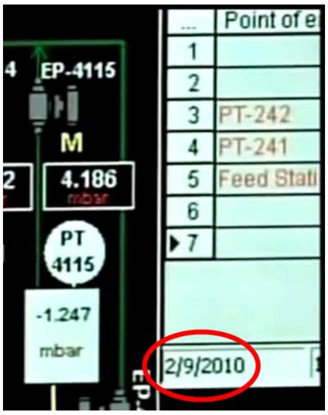
SCADA software heritage

In a facility of strategic importance like Natanz one would expect to find a standard SCADA software package from one of the leading vendors, for example the WinCC software from Siemens. However, such standard COTS product was obviously not used in the control room at Natanz – at least we were unable to spot one. In the screens we have analyzed no vendor logo or other tell-tale indications that would point to a popular SCADA product could be identified.



Screen layout and functionality of the SCADA software appear quite amateurish by Western standards, and crude dialog boxes pop up every now and then on the CPS monitoring application. In modern SCADA software, such pop-up windows are rarely used because they obstruct other information on the display. Also, standard P&ID symbols and labels have not been used consistently, suggesting that the application was custom-built by a corporation or individuals with little familiarity with contemporary SCADA software design.

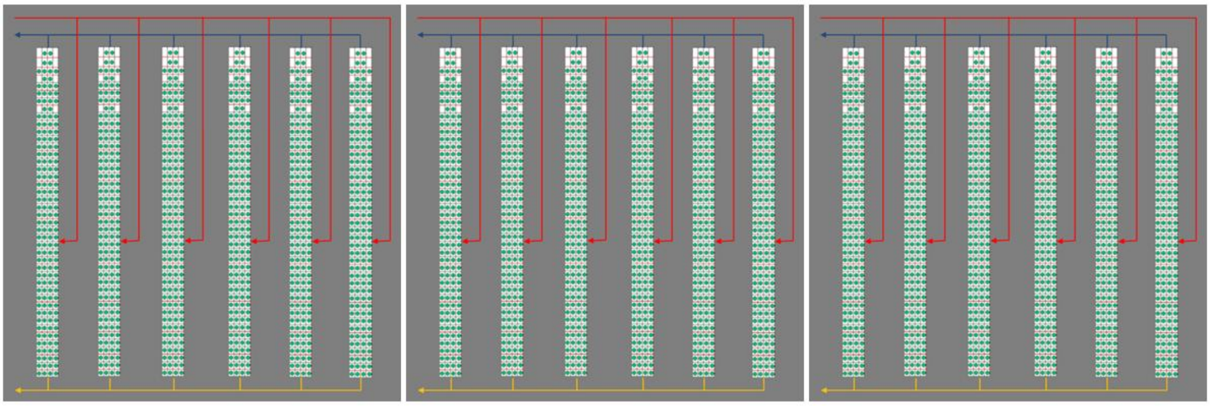
So who developed the SCADA software for Natanz? One would assume that trusted domestic developers were in charge. However, the only Farsi text element we could identify in the screenshots are pretty much unimportant; it appears in the upper left area of the CPS monitor right next to an English language label that seems to read “CASCADE”. Below that label there is a screen area with six push buttons or indicators that are apparently used to switch between the different cascades that make up a cascade unit (the CPS monitors only one cascade at a time). Other labels are consistently in English language. Surprisingly, date is shown US format (MM/DD/YYYY). This screenshot is taken from a video that was shot on February 9, 2010. The information in the text box on the right of the display shows detail information for Pressure Transducer 4110, the pressure sensor for the feed stage, with



the full text in slot 5 most likely reading “Feed Static Pressure”. It appears far-fetched that Iranian engineers would deliberately use the date format of the “Big Satan” unless there is a compelling reason to do so, such as a development team which is very familiar and used to a software development environment with a configuration that is typical for the United States.

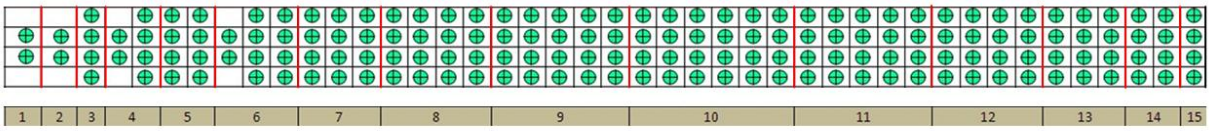
Plant Design

A cascade unit at Natanz is made up of 18 cascades. According to our intelligence, sub-units of six cascades share one feed station, one product station, and one tails station. The Pilot Fuel Enrichment Plant (PFEP) at Natanz also uses six cascades. In the diagram below, red piping indicates feed, blue piping indicates product, and yellow piping indicates tails.



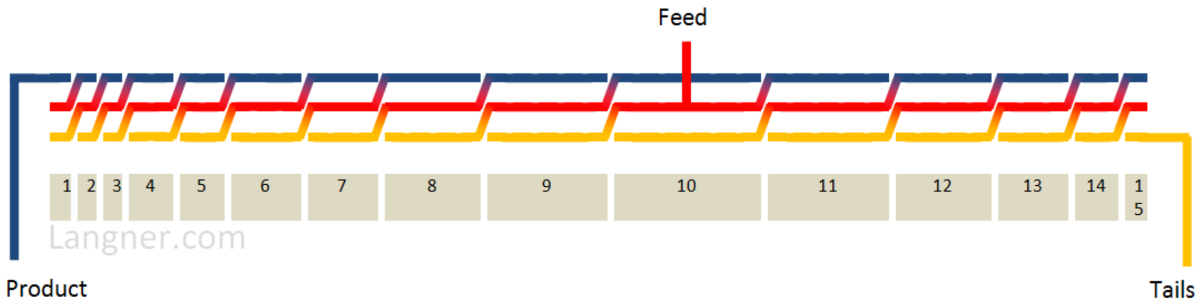
Cascade shape

During the time of the Stuxnet attack (2007-2010), Iran used a cascade layout of 164 first-generation centrifuges (IR-1). Centrifuges are lined up in four lines for a total of 43 columns. For the cascade shape chosen, this design has the benefit that only eight cascade stands need to be left empty.

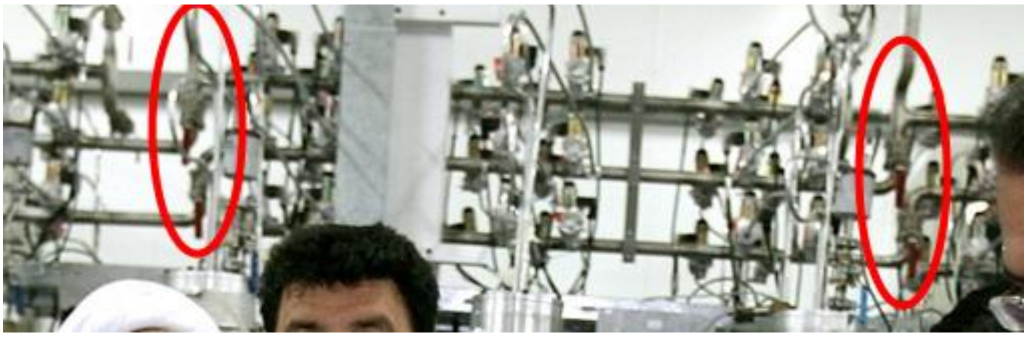


Piping

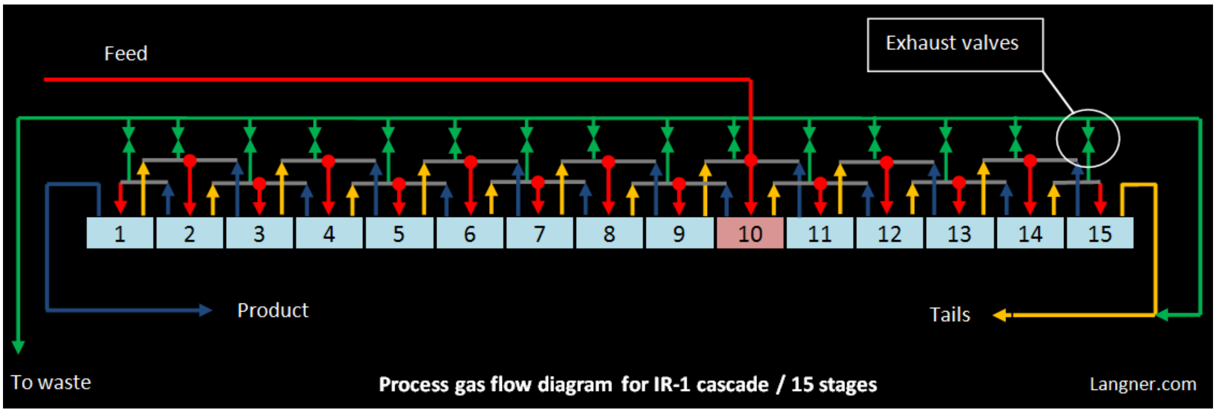
The piping for a cascade is surprisingly simple. Three major pipes are cut-through between enrichment stages, with ends being either welded together or shielded according to the following diagram. Such welded piping is usually referred to as a “fixed configuration”, because the cascade shape cannot be changed without a major pipe job – that would most likely be detected by IAEA inspectors within ample time. In the grey boxes at the bottom, stage numbers are indicated.



The picture below highlights inter-stage connections in the Pilot Fuel Enrichment Plant. It was apparently shot in front of stage 7 or stage 13, which are the only stages in the cascade that are equipped with 12 centrifuges (3x4). The pipes extending to the top of the picture are most likely leading to the exhaust valves and the collective dump pipe.



Process gas flow



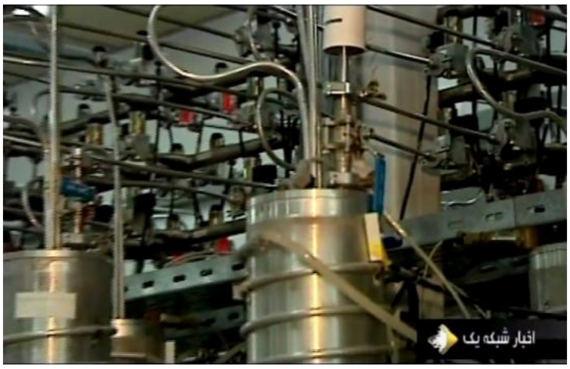
In this diagram, standard I&ID valve symbols are used for stage exhaust valves. Symbols show all exhaust valves open, as would be the case during contingency dump of the whole cascade.

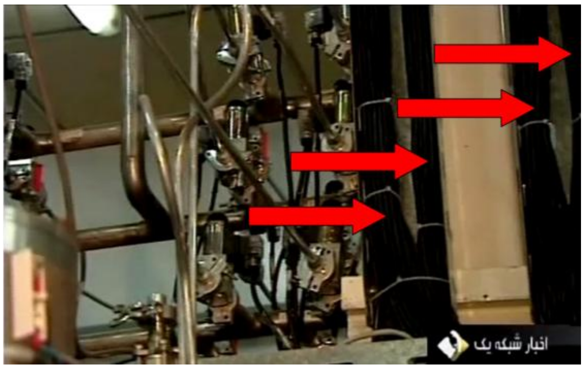
Sensors and Valves

Instrumentation overkill

When comparing an IR-1 cascade with its ultimate predecessor, the original cascade designed and implemented by Urenco, one cannot miss a striking difference at first glance.

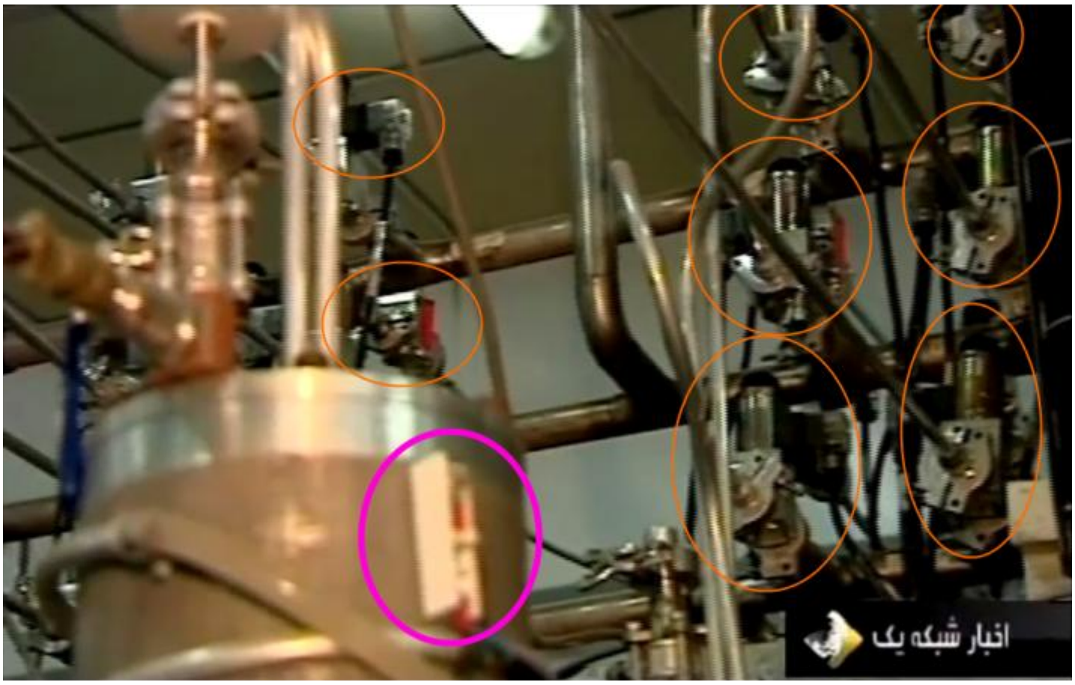
The above picture shows an original Urenco installation. No valves are used, and the cascade isn't cluttered with instrumentation and cabling. Urenco managed to get basically the same design running without a lot of instrumentation and control.





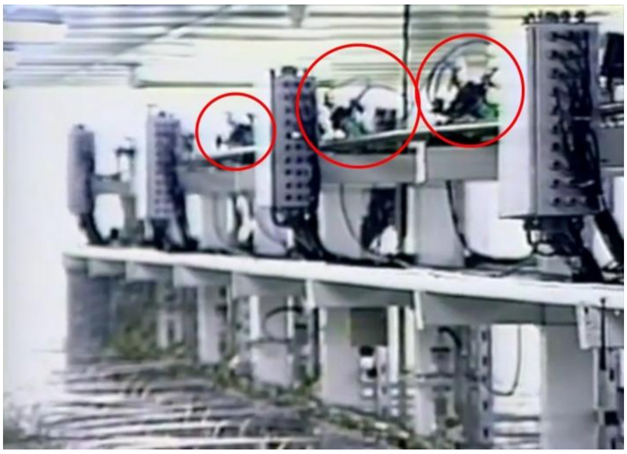
Things are quite different at Natanz. Just a look at the huge signal cable trunks tells that this plant is equipped with a lot of instrumentation that serves one major purpose: Keeping the plant running despite reliability problems. Compared to its Urenco heritage the cascade hall at Natanz looks like an intensive care unit with lots of gear attached to patients in order to keep them alive. Control systems are used to compensate for mechanical unreliability rather than to increase efficiency or product quality.

Centrifuge isolation valves and vibration sensors



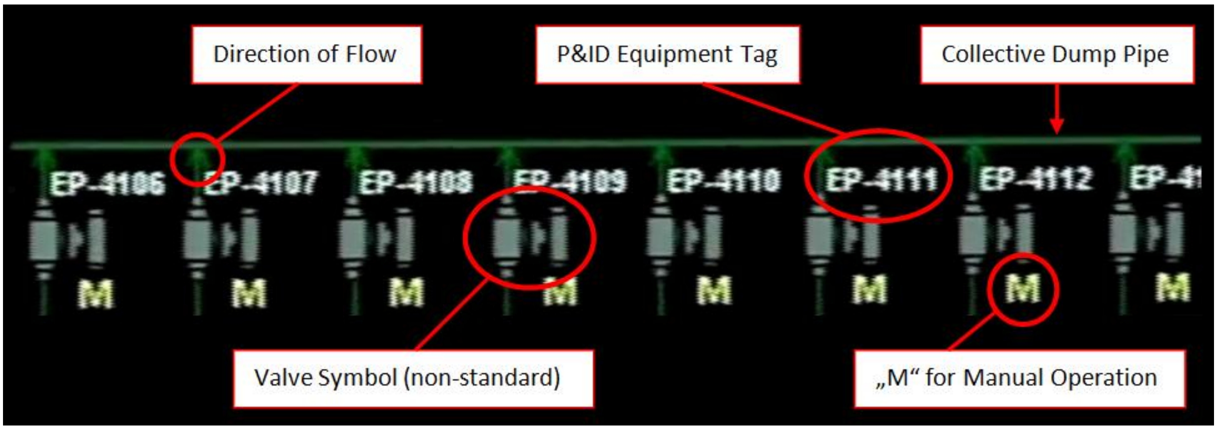
The three connector pipes that connect individual IR-1 centrifuges to the stage feed, product, and tails pipes are equipped with isolation valves, highlighted in orange. The purpose of the valves is to isolate centrifuges from a cascade that start to vibrate, as signaled by vibration sensors (highlighted in magenta). Each valve is connected to a Profibus network attached to Siemens S7-417 controllers.

Stage exhaust valves



Each enrichment stage in a cascade is equipped with a valve through which process pressure can be released into a shared collector pipe which feeds to the dump system. Although there is some uncertainty, we assume that the objects highlighted in red show exhaust valves. Their physical position on top of the cascade and their spacing matches the plant schematics on the SCADA screens.

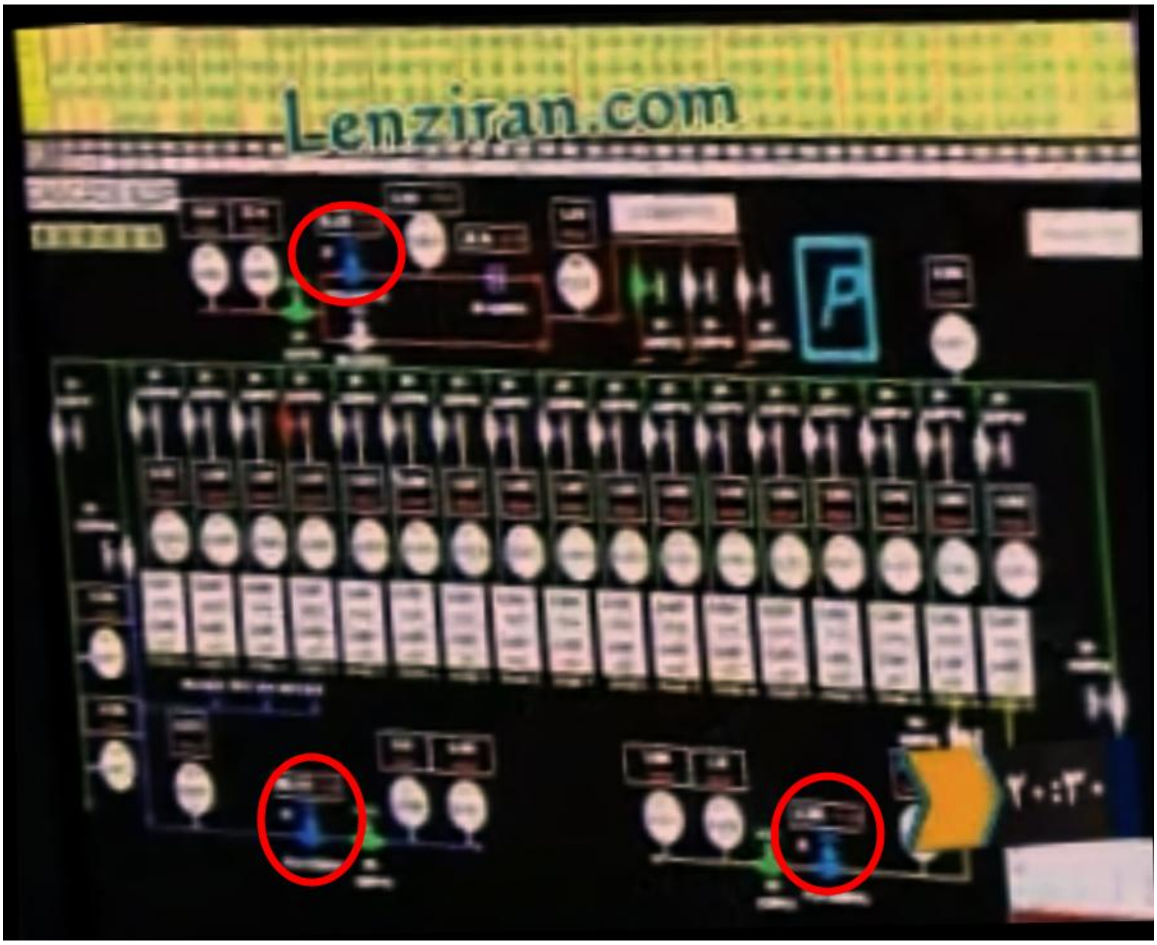
Operation of the valves (open/close) are controlled by an individual pressure controller in respect to pressure sensor readings as discussed below.



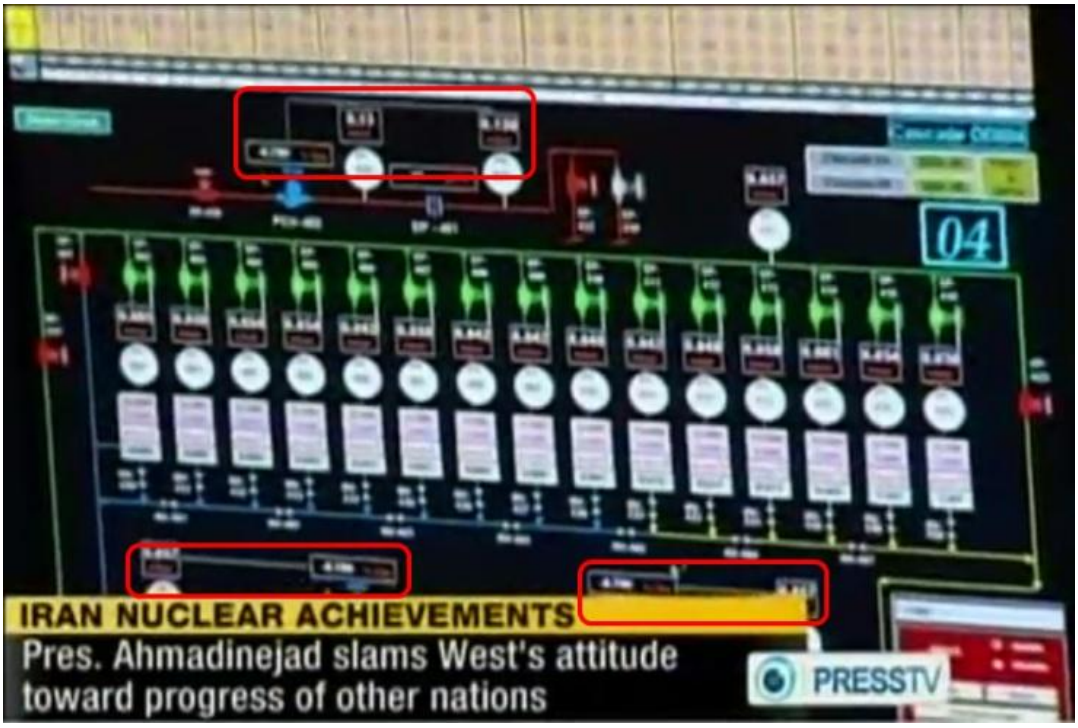
This partial screenshot shows the stage exhaust valves as they appear on the SCADA screens. Each valve is tagged with an identifier starting with “EP-“, which may signify “electro-pneumatic”, the first two digits identifying the cascade, and the last two digits identifying the enrichment stage. The graphical icon used is non-standard; the extension to the right of the symbols may signify a pneumatic pump. The letter “M” beneath each valve apparently stands for manual operation (rather than automatic), where “manual” does not imply an operator actually physically moving a handrail; it stands for an operator manually clicking the mouse in the control room. The screenshot was taken during startup of a cascade, which is usually performed manually.

Control valves

Control valves do not operate binary (open/closed) but can open a pipe to a specific degree. They can be found at the feed, product, and tails headers, at least in the configuration used by Iran since 2012.



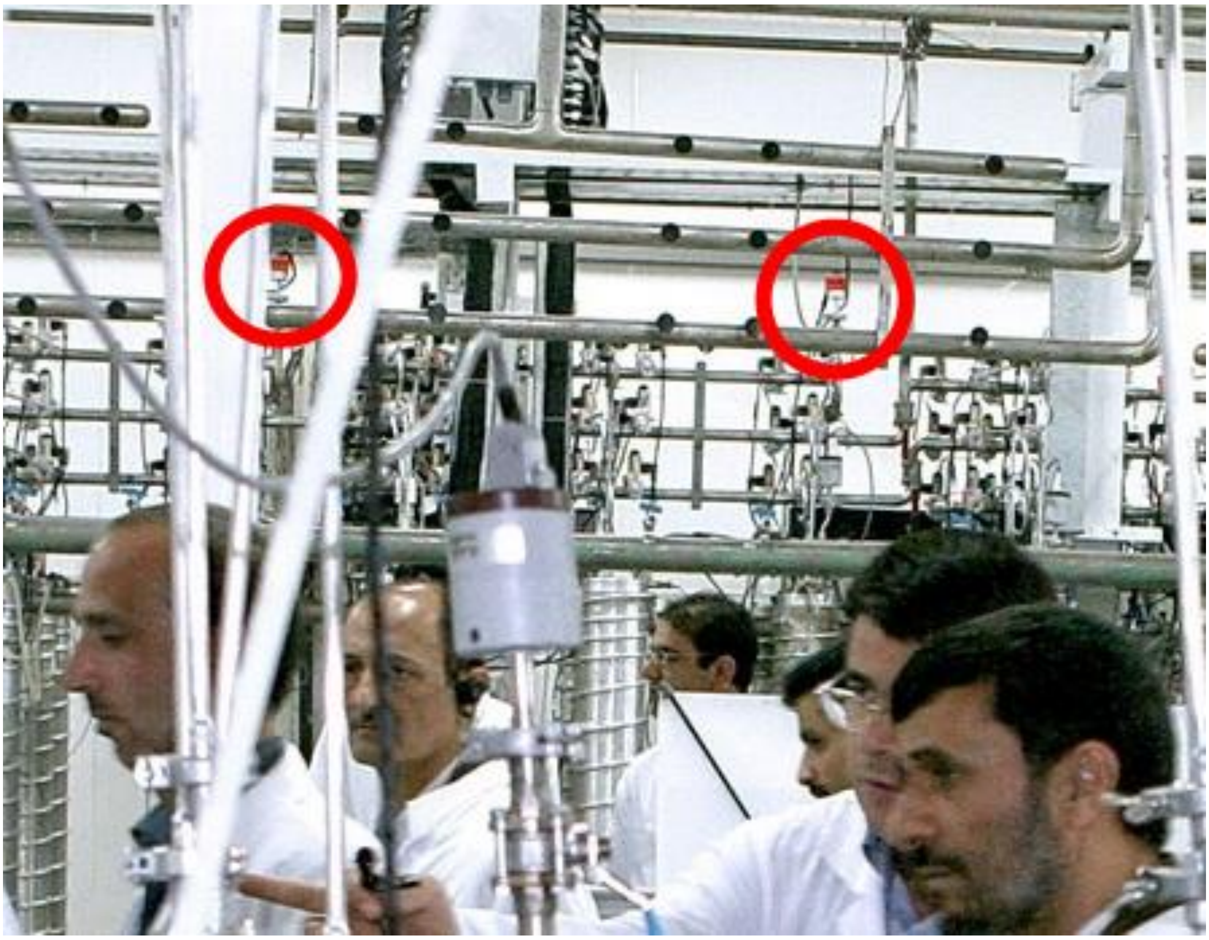
The control valves are obviously operated in respect to the values of the absolute pressure sensors at the feed, product, and tails headers, as the instrument loops that are highlighted in the following picture indicate.


04
IRAN NUCLEAR ACHIEVEMENTS
Pres. Ahmadinejad slams West's attitude toward progress of other nations
PRESSTV

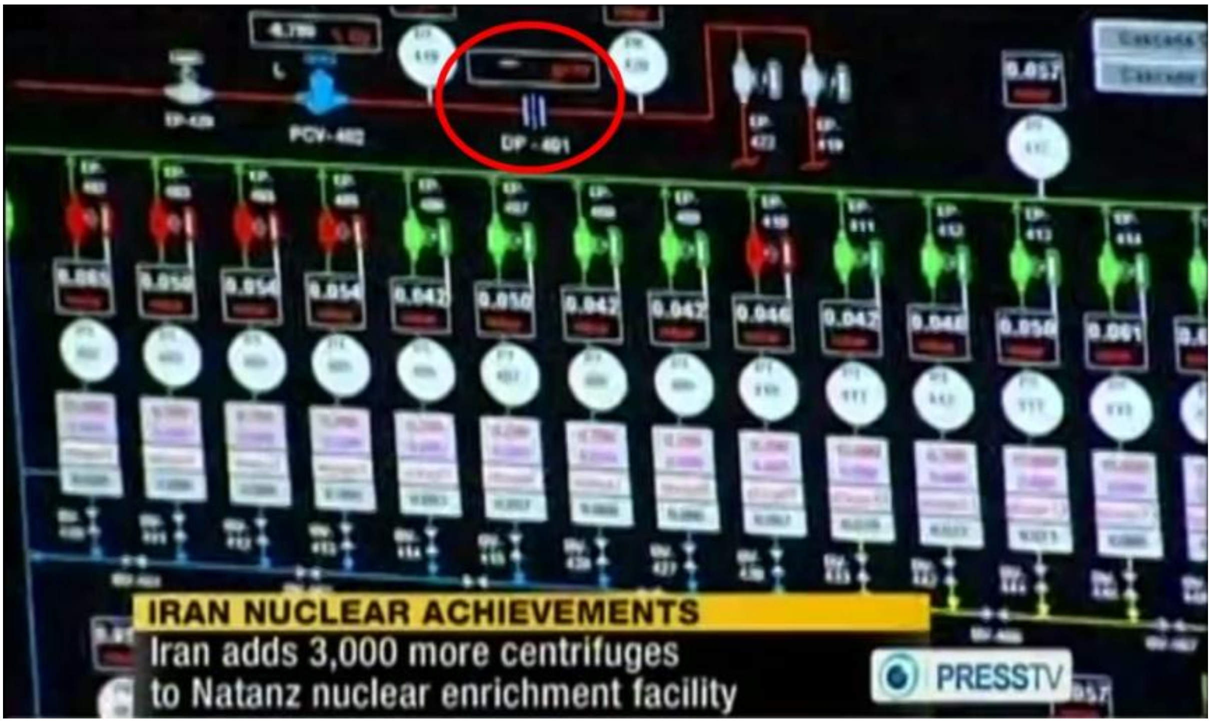
Absolute pressure sensors

Various pictures show the pressure sensors used in Natanz. On the SCADA screens they are labeled with “PT-“, which obviously stands for “Pressure Transducer”. According to our intelligence, Iran uses MKS Baratron sensors, maybe also MKS clones. The following shows an MKS Baratron transducer as photographed by the manufacturer.

Plant floor footage suggests that there are two different groups of pressure sensors: One group that is directly attached to individual centrifuges, and another group attached to stage piping. The following picture shows pressure sensors that appear to be attached to stage piping.



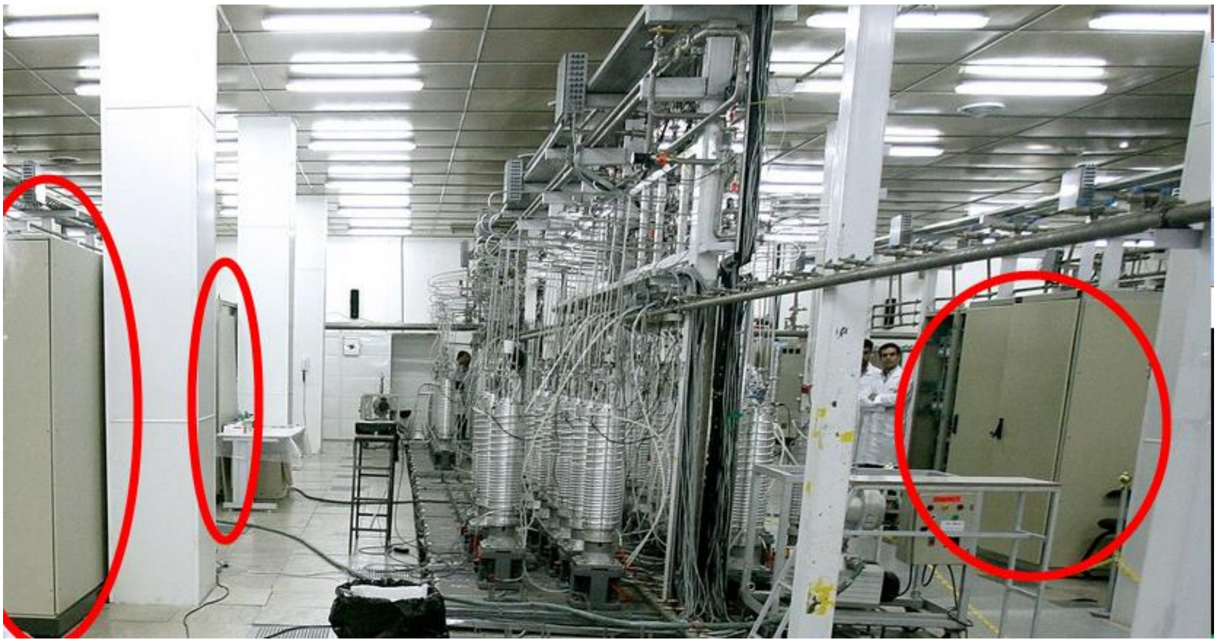
Differential pressure sensors



The only differential pressure sensor that we could identify is located in the feed header in this screenshot from 2012 where it is highlighted in red. It is most likely used as a flow meter.

Industrial Controllers

Control system cabinets



Location of control system cabinets in the above-ground Pilot Fuel Enrichment Plant (PFEP). Although it cannot be seen in the picture, the Siemens S7-417 and S7-315 controllers compromised in the attack are almost definitely located inside these cabinets – which will likely have been accessed directly by the control system engineers who unwittingly infected the controllers with Stuxnet.

Siemens S7-315 and S7-417 controllers

We did not spot any Siemens controllers on plant floor footage, most likely for the simple reason that according to our intelligence, Iran kept the details on their controllers secret, even from IAEA inspectors. Nevertheless it is clear from the attack code that only S7-315 and S7-417 controllers were attacked, with the smaller 315 controlling the Centrifuge Drive System and the larger 417 controlling the Cascade Protection System. Most likely Iran uses the redundant 417H version of the controller to provide for uninterrupted operation in case of controller failure. Software routines dealing with the 417H can be identified in the attack code.

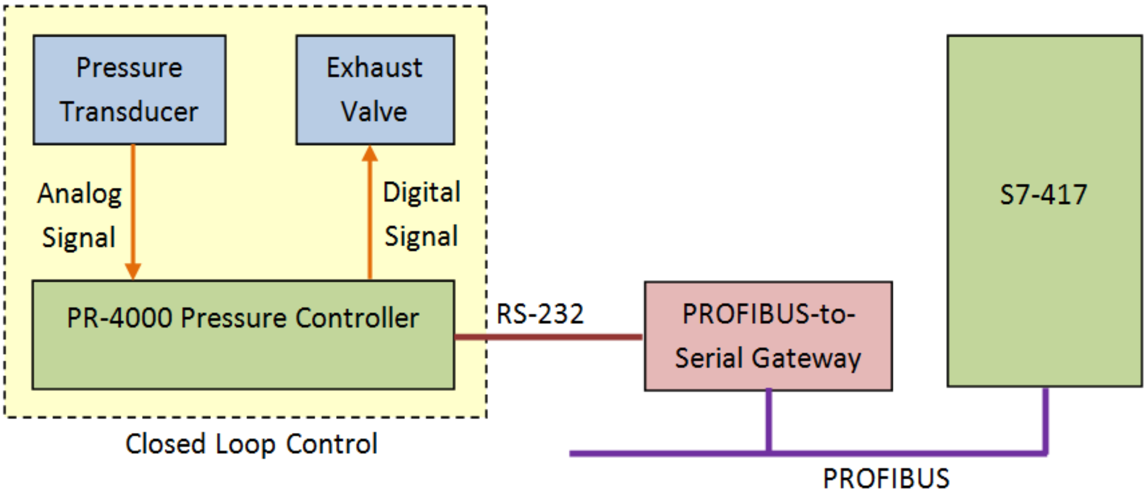
Siemens Field PG

One of the most important things to understand about industrial controllers in respect to cyber security is that they are configured, or programmed, by a mobile computer that runs the vendor's configuration software – a product called SIMATIC Manager. Mobile computers are used because programming usually takes place "in the field", lacking online connectivity to the controllers that need to be configured. The name "PG" is an acronym for "Programmiergerät", which means programming device.

Pressure controller & readout

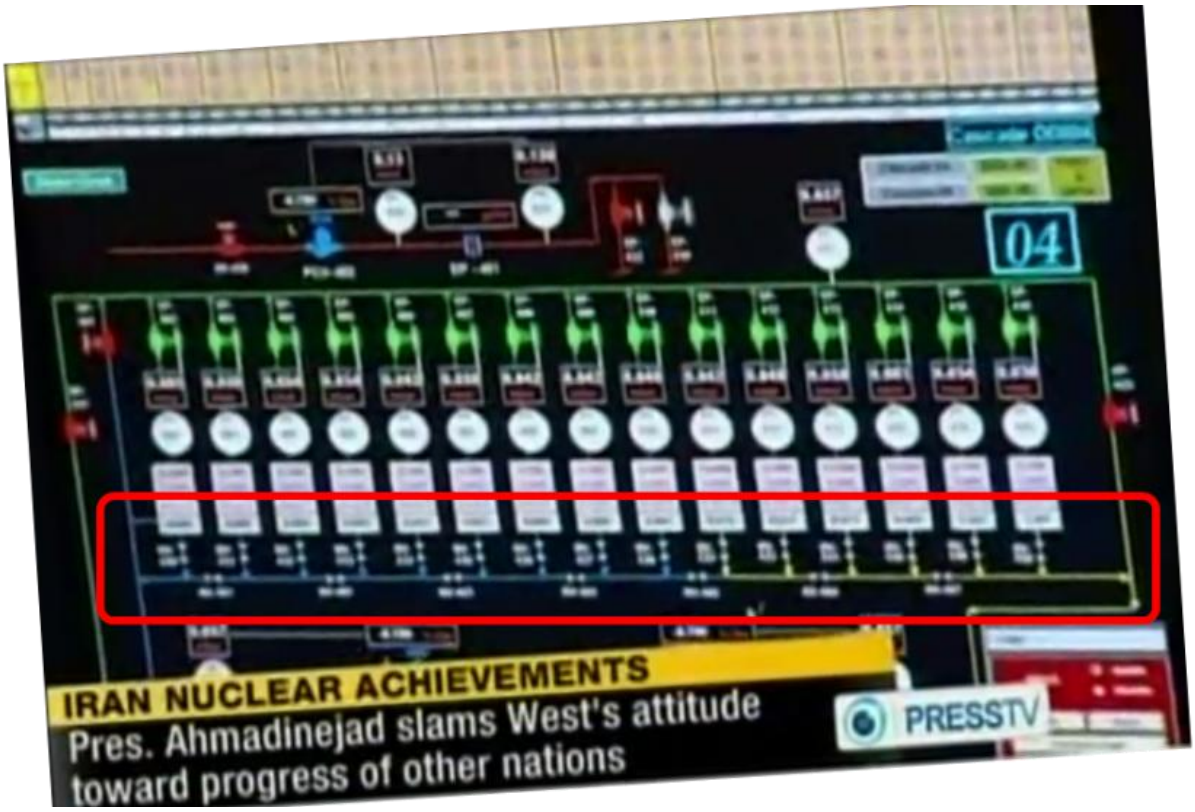
The control units in the pictures above display process pressure and setpoints. The picture below shows the same product (MKS PR-4000) as advertised on the Internet for sale (fergutec.com).

The pressure controllers must be compromised in order to disable the Cascade Protection System's stage exhaust valves. This suggests a link between the Cascade Protection System's main controller, the Siemens S7-417, to the pressure controllers. Since the PR-4000 doesn't come with a built-in PROFIBUS interface, communication is most likely established via a PROFIBUS-to-serial gateway, as shown in the diagram below; a configuration that is used in similar applications. From the attack code it can be inferred that a total of 21 pressure controllers were used per cascade, with the lower 15 controlling stage exhaust valves.



Non-Proliferation Concerns

Analysis of piping, instrumentation and control comes with an unpleasant surprise. A SCADA screen from 2012 indicates that Iran made a move to dynamically configurable cascade profiles.


04
IRAN NUCLEAR ACHIEVEMENTS
Pres. Ahmadinejad slams West's attitude
toward progress of other nations
PRESSTV

The key is the piping *below* the fifteen enrichment stages, highlighted in red. It is equipped with valves that would allow to simply "valve off" leading and trailing stages in order to arrive at a reduced cascade shape with less than fifteen stages. Any other reason for the valves other than to modify the number of enrichment stages is not evident.

Why would one want to reduce the number of enrichment stages? It certainly would be advantageous for the production of weapons-grade uranium. Reduced cascade shapes are used for enrichment levels beyond 20%. For example, Pakistan used cascades with 114 centrifuges to go from 20% to 60% enrichment, and cascades with 64 centrifuges to go from 60% to 90% (weapons-grade) enrichment. While a 164- or 174-centrifuge cascade can theoretically be used to produce weapons-grade HEU, it just takes longer. The smaller cascades largely reduce breakout time. Breakout time is the time a proliferant needs to arrive at nuclear weapons capability after breaking out of the IAEA regime. Another issue that arises is the question if IAEA inspectors had a chance to detect if between their visits cascade configuration has been temporarily changed to produce HEU.

In order to make use of the lesser-stage cascade, the number of centrifuges per stage must be reduced as well. But that can be achieved easily by simply closing isolation valves, as Stuxnet demonstrated (see figure 6).