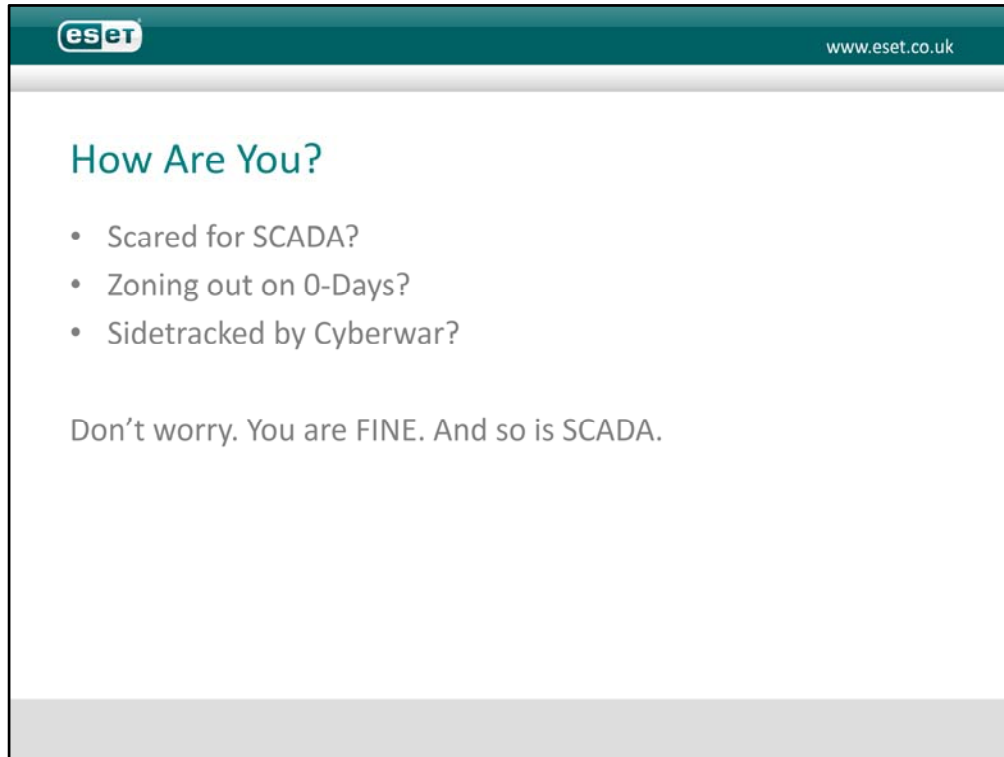




Infrastructure Attacks The Next generation?

David Harley CITP FBCS CISSP
Small Blue-Green World
ESET Senior Research Fellow





So, how are you? Unnerved by all the speculation about Stuxnet, cyberwarfare and cyberterrorism? Don't worry. You're fine.

You know what FINE stands for, don't you? (Thanks for the pointer, Larry!)

Freaked out

Insecure

Neurotic

Emotional

And in case you're unfamiliar with the SCADA jargon...

Acronymically speaking...

SCADA: Supervisory Control And Data Acquisition – coordinates processes

DCS: Distributed Control System – controls processes in real-time

ICS: Industrial Control Systems

CNI: Critical National Infrastructure

RTU: Remote Terminal Unit

PLC: Programmable Logic Controller - cheaper than an RTU

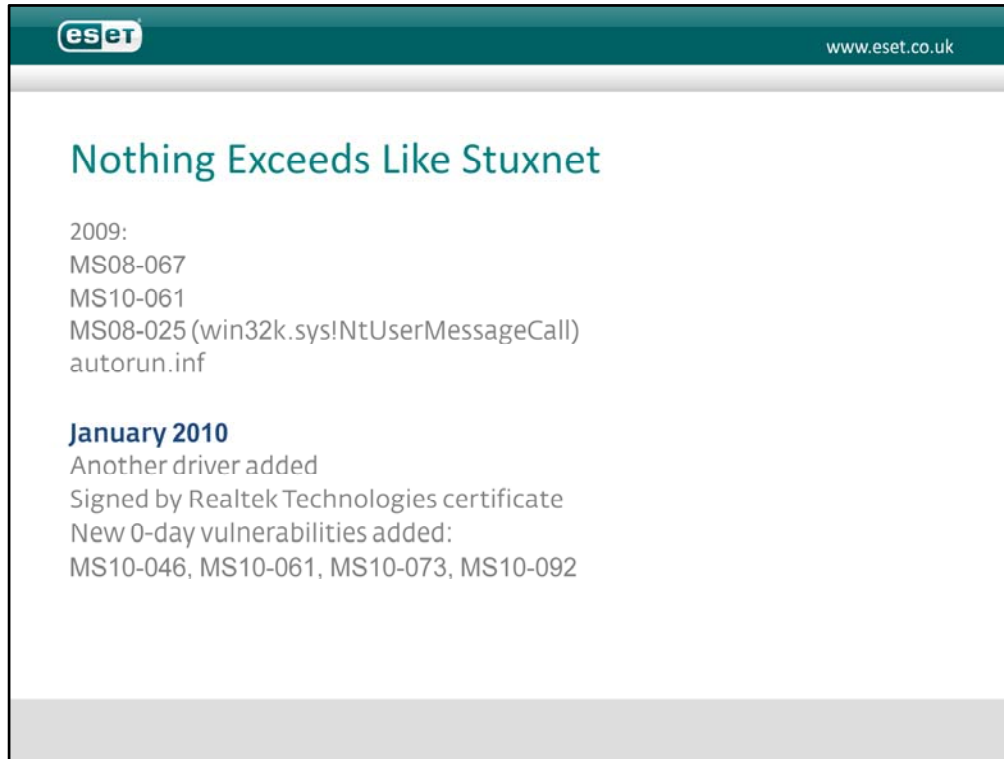
I may make reference to some of these in the course of the presentation, but I'm not focusing on SCADA particularly: I'm more concerned with generic infrastructure as a potential target for malicious action.

What's the Fuss about Stuxnet?

- Unusual Complexity and Sophistication and a FINE array of four 0-days: MS10-046, MS10-061, MS10-073, MS10-092 + MS08-067
- Signed with (stolen) certificates from Realtek and J-Micron
- Tiger team approach to implementation
- Semi-targeted
- Mysterious hardware-specific payload

Stuxnet and its lesser siblings dominated the threat landscape in 2010. Why so much interest?

- An unusually FINE array of multiple exploits of 0-day or little-known vulnerabilities. 0-day, 0-tolerance, and I haven't even listed the Siemens hard-coded "password" fiasco.
- Tiger Team approach to implementation.
- Signed with (stolen) certificates from Realtek and J-Micron. The recent flurry of fake certificates approved by Comodo suggests that signing is an issue that's going to come up again and again. It has its place, but it's not the security panacea.
- It was "targeted", though I'd actually say semi-targeted: I'll come back to that. Still, the payload certainly had a very specific, potentially critical installation in view
- Mysterious hardware-specific payload in a novel control language: certainly a change from Visual Basic and Delphi... And because of the specialized nature of the context in which it executes, it took a long time to ascertain what it actually did, even with specialist help.

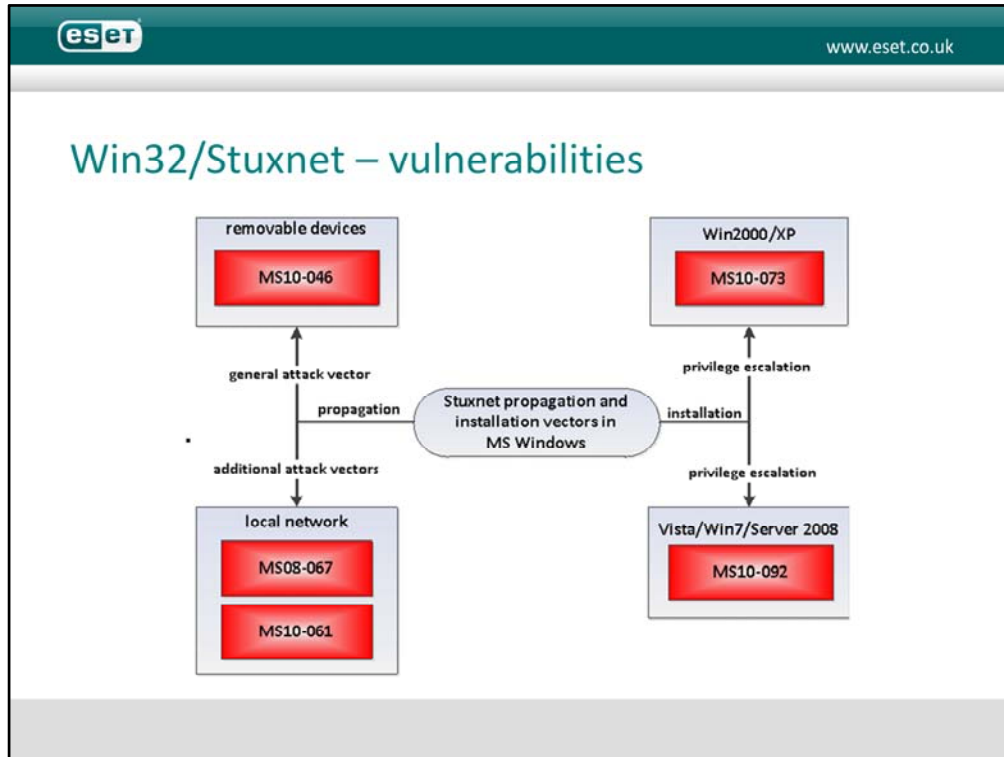


The 2009 version: stayed well under the radar, but in consequence, had limited scope and vulnerabilities. Though the Autorun attack *is* still entirely possible on many systems.

In January 2010, though, there was a significant upgrade:

- Another driver added, signed by Realtek
- Several new 0-day vulnerabilities added

It's unusual to see that many 0-days in a single malicious program. Someone was getting serious.



Let's take a look at one or two of those Liquorice Allsorts 0-days. Multiple entry points, multiple types of vulnerability.

MS10-46 isn't an Autorun infective attack, though it worked very similarly but (for a while, anyway) more effectively - disabling Autorun didn't stop the infection (for that you had to cripple Windows Explorer till the patch came out). Picked up later by a load of short-life bottom feeder malware variants. Any .LNK file could exploit it when displayed in windows explorer and the icon for a .LNK file was loaded from a Windows Control Panel file (actually a DLL, effectively).

MS08-067: self-distribution over the network installing a dropper through shared folders. It scanned network shares c\$ and admin\$ on the remote computers and installed a dropper there with the name *DEFRAG<GetTickCount>.TMP*, scheduling a task to be executed on the next day

(MS10-061). A privilege escalation vulnerability in Window Spooler allowing a remote Guest account to write into %SYSTEM% directory. Machines with file and printer sharing turned on were vulnerable to the attack.

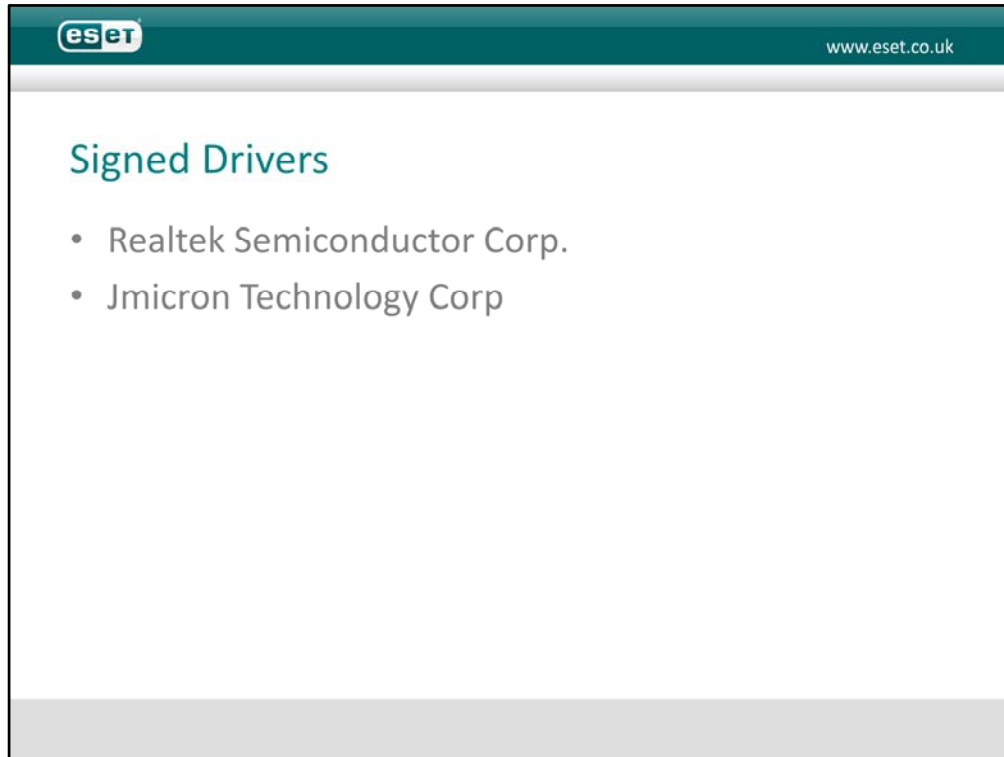
(MS10-073) 0-day in Win32k.sys: Escalated privilege level up to SYSTEM, which enabled it to perform any tasks it likes on the local machine. To perform this trick, Stuxnet load a specially crafted keyboard layout file, making it possible to execute arbitrary code with SYSTEM privileges.

MS10-092 attack used a vulnerability in the Task Scheduler service to escalate privilege: in fact, that attack lives on the TLD4 bootkit. TLD4's implementation doesn't essentially differ from that of Stuxnet's code. The rootkit created a legitimate task by means of the available interface in the system, then read the xml schema corresponding to the task directly from the file in the Task Scheduler folder, and then modified it.

MS10-046 related malware and its evolution

Malware name	first appearance ITW	LNK exploit added	signed	advanced	prevalence	targeted
LNK/Exploit.CVE-2010-2568	2010/07/16 (2008/11/20)	2008/11/20	N/A			N/A
Win32/Stuxnet	2009/01	2010/03 (2010/01?)				
Win32/Autorun.VB.{RL, RP, RT, RU, SN}	2010/07/18	2010/07/22				
Win32/Sality.NBA	2003/07/06	2010/07/24				
Win32/Agent.OTB	2010/01	2010/07/26				
Win32/TrojanDownloader.Chymine.A	2010/07/13	2010/07/26				
Win32/Delf.NVR (xbot)	2010/07/09	2010/07/27				
CN "0-day"	2010/08/02	2010/08/02				
Win32/Agent.OSW aka Dottun (fanny)	2008/07	N/A				

But more is not always more. Here are some of those bottom feeders.



Both companies whose code signing certificates were used had offices in Hsinchu Science Park, Taiwan. That may suggest physical theft, but certificates may have been bought from another source. For instance, the Zeus botnet is known to steal certificates, though it focuses on banking certificates.

The file `jmidebs.sys` functions in much the same way as the earlier system drivers, injecting code into processes running on an infected machine. Maybe the attackers changed their certificate because the first one was exposed, or were simply using different certificates for different attacks. Either way, they obviously had significant resources to draw on. The modular architecture that characterizes the Stuxnet malware, suggests the involvement of a fairly large and well-organized group.

Easy, Tiger

- The anti-Tiger team: coalition of entities with specialist expertise.
- Yet the (later) version we became aware of was promiscuously distributed.
- Why?

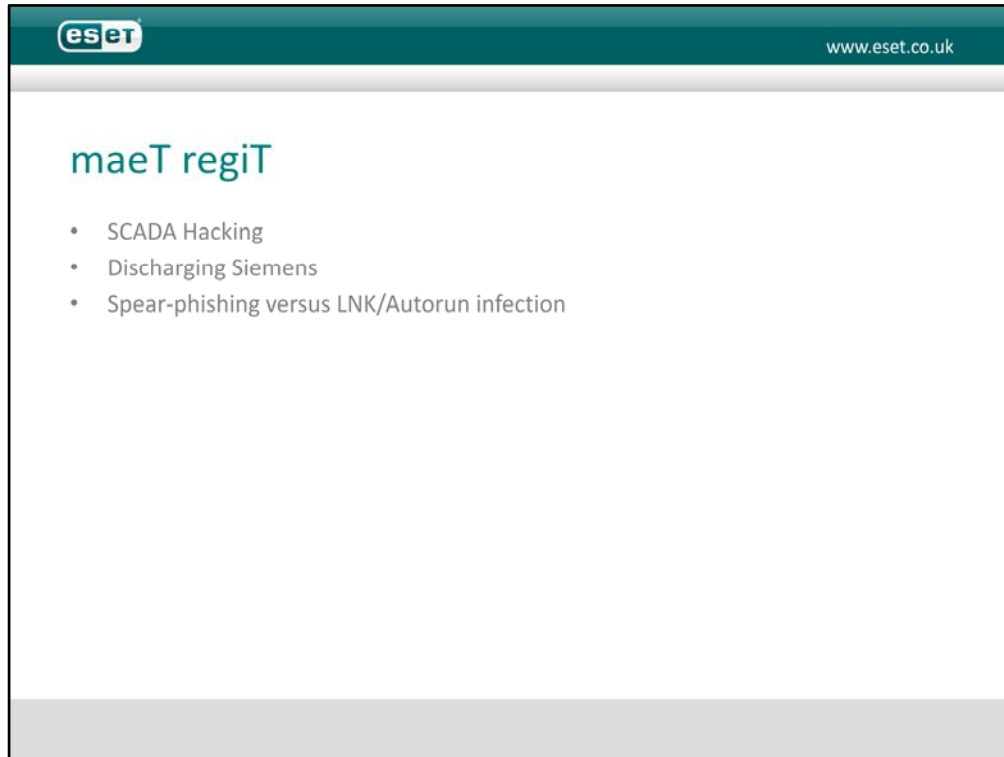
Stuxnet has the hallmarks of a collaboration between several individuals or groups with specialist expertise, yet its cover was blown by its promiscuous dissemination through the Autorun-like LNK vulnerability, a vector that automatically raised its chances of being detected heuristically. That suggests to me...

*Tyger, Tyger...



- A team where no-one had specific experience in the malware field (maybe that's understandable in a team put together under the auspices of a government or governments, conspiracy theories notwithstanding).
- Or the malware had already been so effective that staying under the radar wasn't a major concern (the LNK version of Stuxnet was not the *first* version).
- Or someone intended to send a message to Iran and the rest of the world about the capabilities of certain agencies and states. After all, much has been made of the "clues" in the code. However, I still haven't seen conclusive proof that it was the US and/or Israel that orchestrated that "message", or planted those clues. If either of those nations are really hinting that they did, that doesn't make it so: misdirection is a standard tool for diplomats and politicians, as well as for spooks.

* http://en.wikipedia.org/wiki/The_Tyger

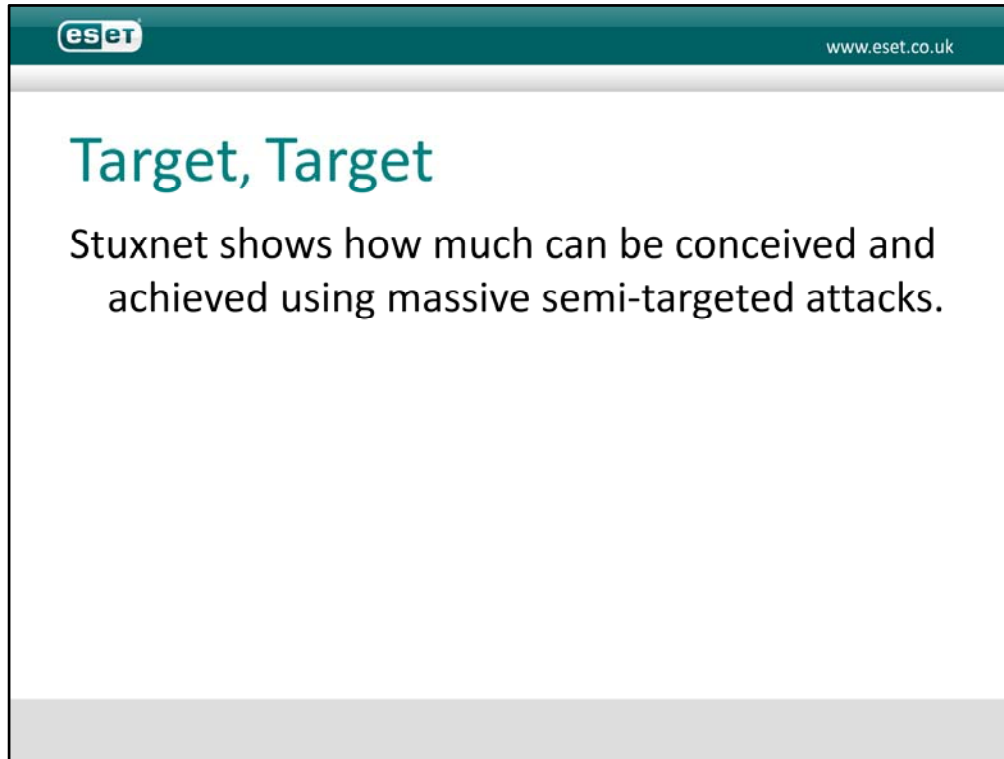


No, I'm not exercising my rusty Welsh: that's a play on the idea of a Reverse Tiger Team. Ironically, my initial involvement with Stuxnet is partly due to being pulled into a Tiger Team in the US incorporating SCADA sites, law-enforcement, and one or two security vendors.

Anyway, some of the code looks as if it originated with a "regular" software developer with extensive knowledge of SCADA systems and/or Siemens control systems, or a formally-constituted, multi-disciplinary "tiger team". It looks less like the work of the criminal gangs responsible for most malware, or even the freelance hacker groups sometimes funded by governments and the military (for example Wicked Rose) we often associate with fully targeted attacks.

On the other hand, wide propagation prevented the malware from staying "below the radar". This may signify misjudgement or a miscommunication at some point in the development process.

On yet another hand (I like to carry a spare), it may simply mean that the group was familiar enough with the modus operandi characteristic of SCADA sites to gamble that Stuxnet would hit enough poorly-defended, poorly-patched and poorly-regulated PLCs to gain them the information and control they wanted. The clustering of infections in a country characterized by software import restrictions and patchy observance of licensing requirements may be significant.



...burning bright...

Why semi-targeted? While the payload is plainly focused on a control system, the malware's propagation is promiscuous. Criminal (and nation-state funded) malware developers have generally moved away from the use of self-replicating malware towards Trojans spread by other means (spammed URLs, PDFs and Microsoft Office documents compromised with 0-day exploits, and so on). Truly targeted non-replicating malware (aimed at individuals, often using customized social engineering as well as customized code) is much harder to catch.

Stuxnet's usefulness in terms of payload delivery may well have been depleted (little nuclear joke there) by public awareness of the threat and the wider availability of protection.

What's the real game-changer?

What's the real game changer?

- Potential targets
- State of SCADA security before Stuxnet
- State of SCADA security after
- What the wider business community can learn

But what are the real implications behind the hypefest and the barrage of sheer speculation about who created it, and why? The real game-changer here is not the malware, interesting though it is to specialists and the media alike, but what it says about its potential targets and the state of security before and after.

The image is a screenshot of a webpage from ESET. At the top left is the ESET logo, and at the top right is the URL 'www.eset.co.uk'. The main heading is 'It's really not about Stuxnet'. Below the heading, the text reads: 'Not the malware...', 'Not the origin and targeting...', and 'Not even the painstaking binary analysis...'. This is followed by 'It's about:' and 'National/International state of SCADA security'. Below that is 'Security in a world of cyber-everything:' followed by a bulleted list: 'Cyberespionage', 'Cyberterrorism', 'Cyberwarfare', and 'Cyberhysteria'. The page has a dark teal header and a light grey footer.

It's not the speculation about origin and targeting of the malware, or even the painstaking analysis of the binaries, but the way in which critical installations work, and their relationship with the wider world of security. What does it tell us about the state of CNI (Critical National Infrastructure) security, nationally and globally? Where does the security industry fit in a world of cyber-everything?

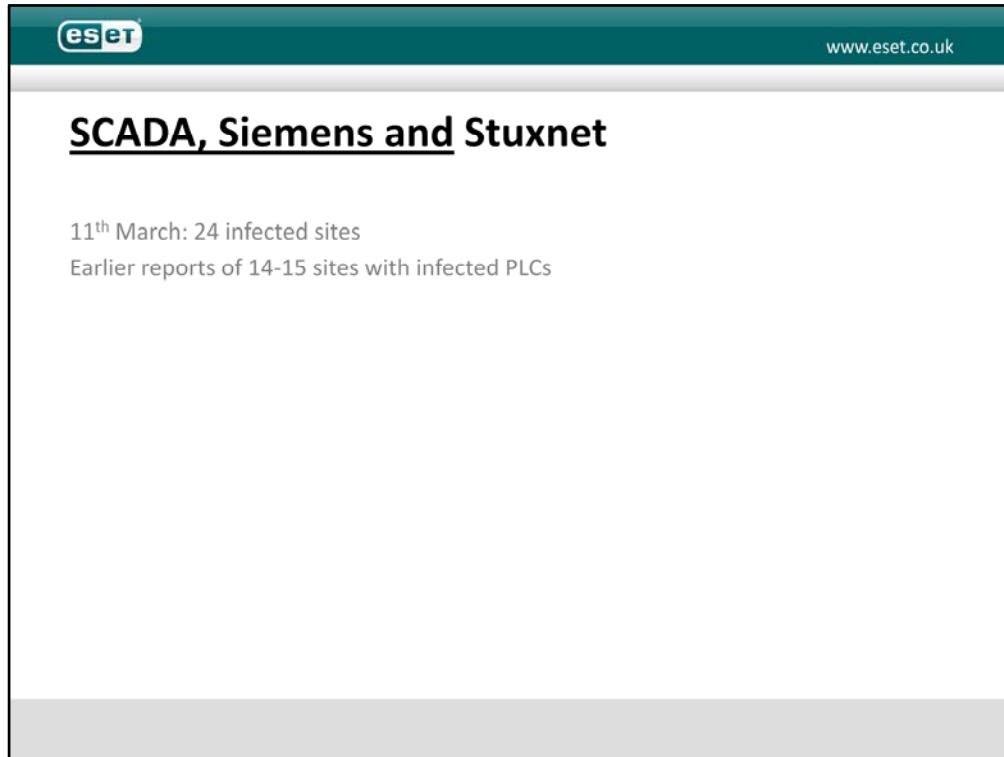


Here's a selection of suggestive incidents from New Scientist:

- Byzantine Candor 2002-2007, targeting military and government agencies in the US: "Massive amounts of sensitive data stolen".
 - Ghostnet, 2007-2009. Multiple targets, but everyone associates it with attacks on India's embassy in the US and the offices of the Tibetan government in exile. Exact nature of the damage done unknown, but system resources clearly infiltrated.
 - Aurora, 2009. Targeted Chinese human rights activists and some big players in the US technology industries, notably Google. Proprietary code stolen, activist emails compromised.
 - Shadows in the Cloud, 2009-2010: Targeted Indian and Tibetan government offices and the United Nations. Sensitive correspondence and documents compromised. We could add some other stuff here, like the Night Dragon attacks on petrochemical companies like Exxon, Shell and BP.
- Attacks from Russia on Estonia and Georgia, targeting a range of web sites including government, media and finance organizations.
- Wikileaks. Well, that's a can of worms I'll keep for another time, another place.
- Stuxnet: allegedly an attack by Israel and the US on Iran (oh, really?)

There's other stuff they could have mentioned:

- Titan Rain: alleged theft of military Intel by China on (Lockheed, NASA, Sandia)
- Moonlight Maze: also targeted military intel, allegedly, by Russia (Pentagon, NASA, Dept of Energy, research labs)
- The Siberian pipeline explosion in 1982, alleged to have been caused by a CIA logic bomb, but the truth of that story has been debated. I'm not even going to mention the Desert Storm printer virus. (OK, I did, but I'm not going to discuss it.)

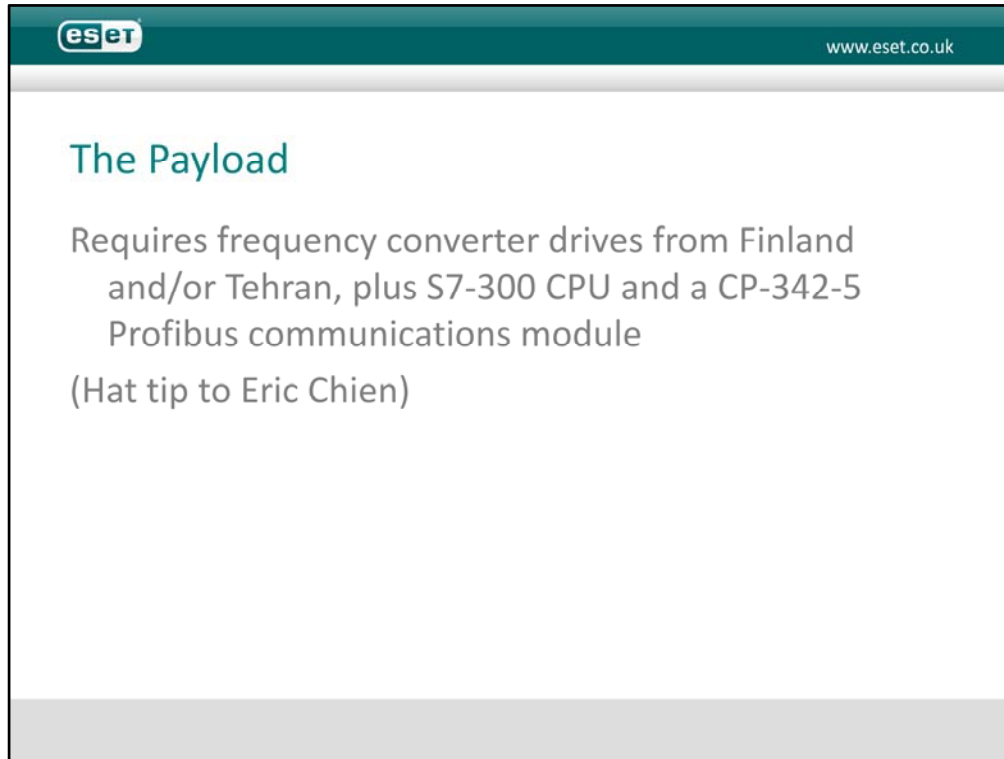


As of 11th March, Siemens was reporting 24 industrial sites with (unspecified) infected systems

There were earlier reports of 14 or 15 SCADA sites affected by the direct infection of PLCs (Programmable Logic Controllers). While the use of these vectors has increased the visibility of the threat, it's likely that it has also enabled access to sites where "air-gapped" generic defences were prioritized over automated technical defences like anti-virus, and less automated system updating and patching. This is not a minor consideration, since the withdrawal of support from Windows versions earlier than Windows XP SP3. One thing that became very clear was that Microsoft was not rushing to patch unsupported systems so as to save necks at SCADA utilities using obsolete Windows versions. At the same time, it's clear that there are difficulties for some sites where protective measures may involve taking critical systems offline. While there are obvious concerns here concerning SPoFs (single points of failure), the potential problems associated with fixing such issues retrospectively should not be underestimated.

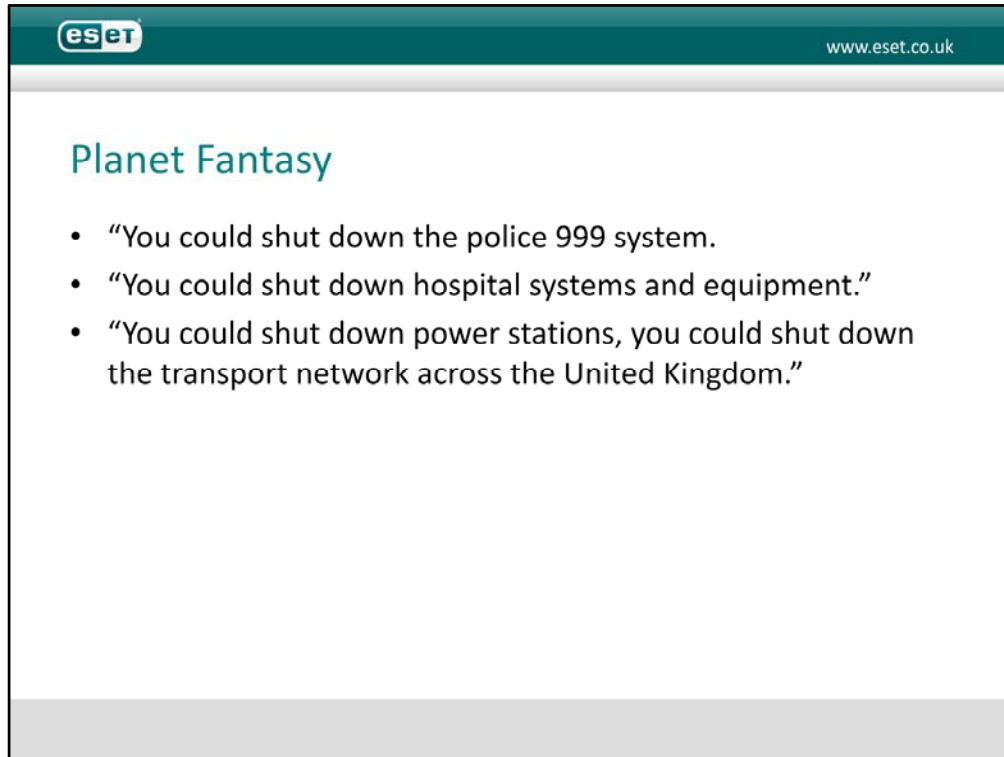
This attack made additional use of a further vulnerability categorized as CVE-2010-2772, relating to the use of a hard-coded password in those systems allowing a local user to access a back-end database and gain privileged access to the system. This meant not only that the password was exposed to an attacker through reverse engineering, but, in this case, that the system would not continue to work if the password was changed, though that issue was not mentioned in Siemens' advice to its customers at <http://support.automation.siemens.com/WW/view/en/43876783>.

Industrial Controls Engineer Jake Brodsky made some very pertinent comments in response to David Harley's blog at <http://blog.eset.com/2010/07/20/theres-passwording-and-theres-security>. While agreeing that strategically, Siemens were misguided to keep hardcoding the same access account and password into the products in question, and naive in expecting those details to stay secret, Jake pointed out, perfectly reasonably, that tactically, it would be impractical for many sites to take appropriate remedial measures without a great deal of preparation, recognizing that a critical system can't be taken down without implementing interim maintenance measures. He suggested, therefore, that isolation of affected systems from the network was likely to be a better short-term measure, combined with the interim measures suggested by Microsoft for working around the .LNK and .PIF issues that were causing concern at the time (<http://support.microsoft.com/kb/2286198>).



This is the intended payload, give or take some detail, as eventually described by Symantec.

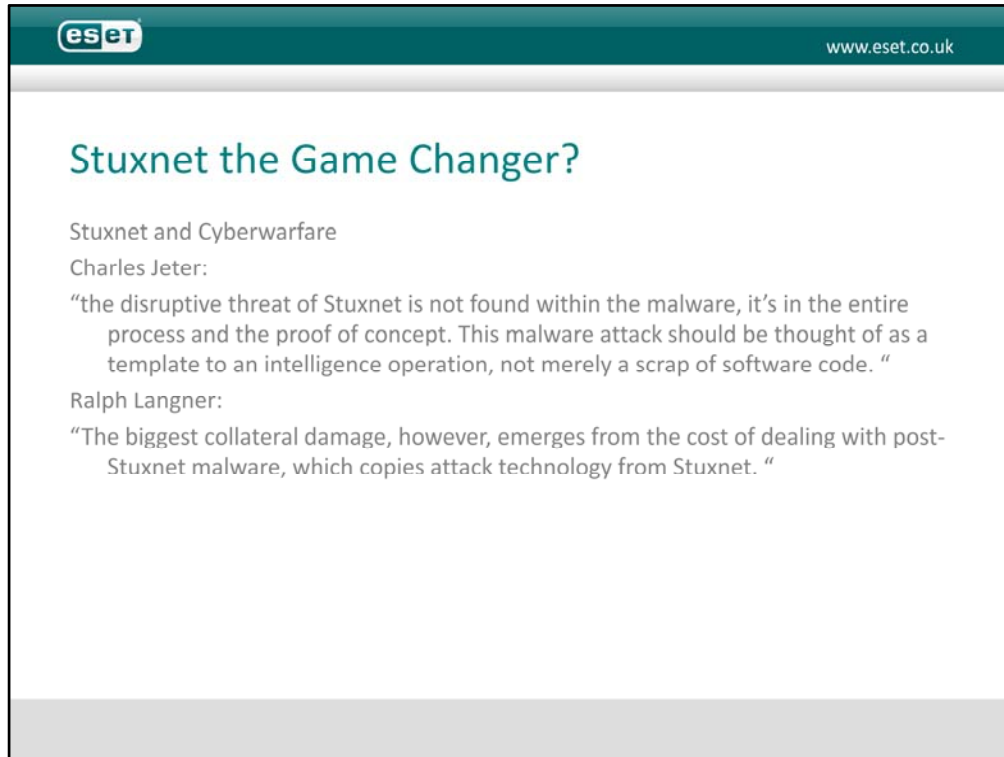
There is certainly substantial evidence suggesting that equipment used for uranium enrichment in nuclear facilities, perhaps in Iran, was the original target.



This didn't put a complete end to the speculation, of course. In fact, some of the speculation actually grew wilder. Most notably, Sky News discovered that the Sky really is falling, claiming that the "super virus" is being traded on the black market and "could be used by terrorists" (presumably as opposed to the saintly individuals who originally put Stuxnet together, apparently to attack nuclear facilities).

In fact, given the amount of detailed analysis that's already available, anyone with malicious intent and a smidgen of technical skill would not need the original code. There is certainly substantial evidence suggesting that equipment used for uranium enrichment in nuclear facilities, perhaps in Iran, was the original target. However, Will Gilpin, apparently an IT security consultant to the UK government, suggested that possession of "the virus" in whatever form has alarming potential.

The assertions above clearly owed little to the PLC code actually discussed in the competent analyses above. While it might be possible to do all these things, that would require extensive re-engineering of the existing code and possibly a completely new set of 0-days.



Stuxnet: Cyberwarfare's Universal Adaptor?,

Charles Jeter (formerly a contributing writer for the ESET blog and Securing Our eCity): "the disruptive threat of Stuxnet is not found within the malware, it's in the entire process and the proof of concept. This malware attack should be thought of as a template to an intelligence operation, not merely a scrap of software code."

<http://blog.eset.com/2010/10/14/stuxnet-cyberwarfares-universal-adaptor>

Ralph Langner: "The biggest collateral damage, however, emerges from the cost of dealing with post-Stuxnet malware, which copies attack technology from Stuxnet. "

<http://www.langner.com/en/index.htm>

Impact on the Security Industry

- Spread and Detection
- Targeted malware versus targeted payload
- Disentangling the payload
- Detection: specific versus generic versus proactive

What has the impact of Stuxnet been on the IT security industry?

In terms of detection, it ceased to be much of a problem once it was widespread enough for us to know it was there. Earlier versions stayed under the radar for many months because its distribution was extremely limited and pretty much confined to an industry sector where tight AV protection is not always seen as necessary or even possible. (Iran seems to have had particular problems arising from software import restrictions and a somewhat lax attitude to licensing.) But the versions that hit the headlines, though often described as targeted, were actually targeted in terms of the payload (what it did and where it activated), much less so in terms of distribution. Hence the initially wide geographical dispersion of reported infections, though the clustering in Iran turned out to be highly suggestive.

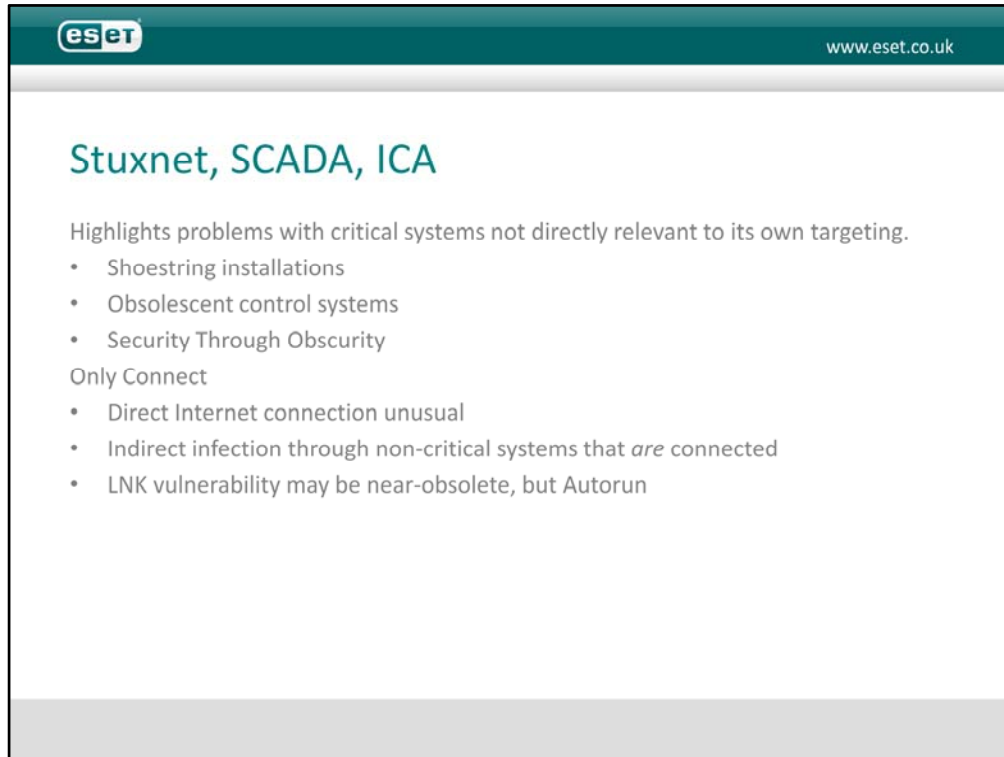
For AV, the real challenges were in disentangling exactly what it did. Fortunately, you don't have to know every detail of the payload to detect malware, but the complexity of the attack made detailed analysis very time consuming, where it was possible at all. Most AV companies don't have ready access to specialist SCADA systems, or the specialist expertise to analyse the PLC code. And in any event the code doesn't tell you everything about the target. At ESET, we concentrated more on analysing the 0-day attacks, from which we learned a great deal. This is important because so did the bad guys. For instance, the TDL4 bootkit uses the MS10-092 Task Scheduler exploit we first encountered in Stuxnet. (We just published a paper on that.) While close analysis of such exploits isn't usually necessary in order to detect specific malware, it enables us to do a better job of developing generic and proactive detections that lessen the impact of future malware that uses similar attack techniques.

Security is an ongoing process

- Continuous review of security posture and risk assessment
- Evaluation of and adaptation to new risks (internal and external)

Do organisations and governments need to re-evaluate their security?

Constantly. Not just because of Stuxnet – the threat from direct re-use of Stuxnet SCADA-specific code in unrelated attacks (as opposed to more generic re-usable binary techniques) has been exaggerated – but because sound security isn't just a one-off implementation: it's an ongoing process to which continuous review of security posture and risk assessment are integral. Evaluating and adapting (where necessary) to new risks, internal and external, is part of the process.



What does Stuxnet mean for the computer security and control systems industry?

There are difficulties for some sites where protective measures may involve taking critical systems offline. While there are obvious concerns here concerning SPoFs (single points of failure), the potential problems associated with fixing such issues retrospectively should not be underestimated.

Stuxnet brought to light some problems with SCADA and ICS that aren't necessarily directly relevant to Stuxnet's own targeting. There are some scarily bright people in that space who understand both security and their own specialism very well, but there are also a lot of cut-rate installations relying on obsolescent control systems and security by obscurity. A critical installation doesn't usually have direct connection to the Internet, but infection is possible via connected systems that aren't in themselves critical. Think USB, for a start.

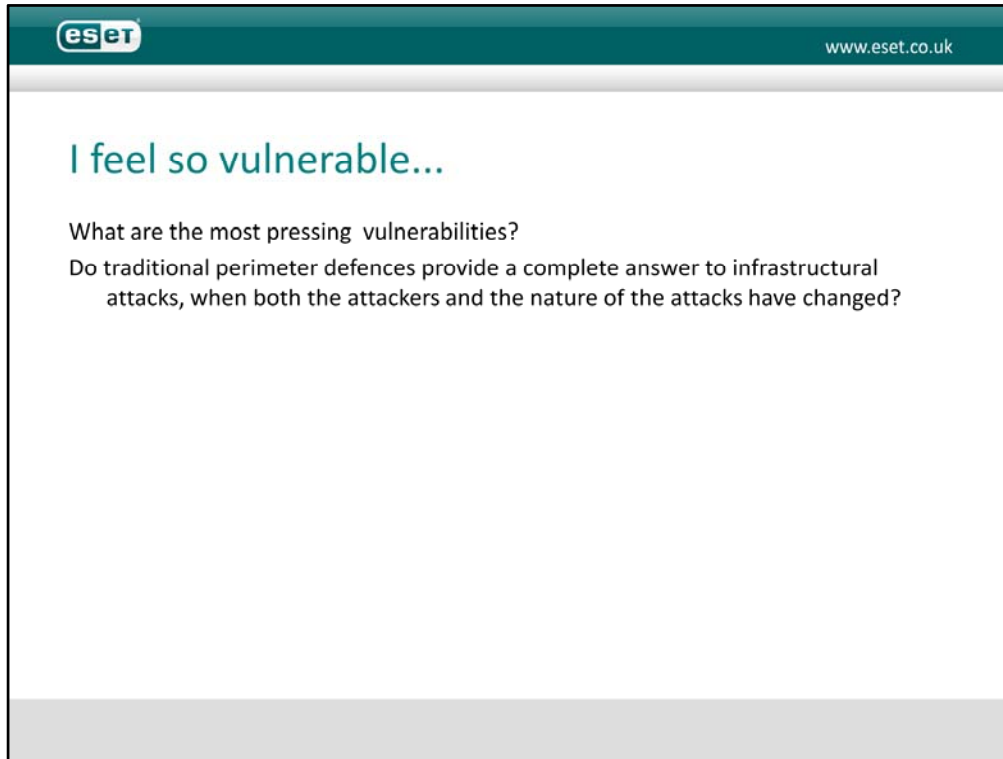
At this time, while the latest implementations of Windows have addressed the problems with Autorun infections via USB devices (though they haven't eliminated them), they account for a huge proportion of the malware ESET detects. The generic INF/Autorun detection was, once again, top of the top ten list for ESET's April 2011 report based on ThreatSense® telemetry.

How has it affected governments and the military?

There's more to this than powerplants

- Multi-disciplinary teams rather than samurai
- Potentially critical sites attacked
- Political and military implications

The kind of utility that Stuxnet appears to target is important, but there's a lot more to a critical national infrastructure than nuclear reactors. The message is not so much that a nuclear programme can be derailed by malware (though it remains a subject for debate as to how true this was of Bushehr or Natanz) but that somebody thinks it's worthwhile to put together teams to orchestrate attacks on critical sites using malware, rather than just hiring a few samurai to launch targeted attacks, as happened a great deal in the early noughties. The next such attack may be focused on a very different sector and use entirely different exploits and attack vectors, but it's unlikely that there will never be another attack of this type. Political and military strategists cannot ignore that potential.



Where are the most pressing vulnerabilities?

All the Microsoft vulnerabilities have been addressed, though in some cases it took a loooooooooooooooooong time. There are still grounds for concern over past and future issues with Siemens. ICS-CERT's April monitoring report flags other, ongoing SCADA issues with default logon credentials

Do traditional perimeter defences provide a complete answer to infrastructural attacks, when both the attackers and the nature of the attacks have changed?



This is a screenshot of a genie emerging from a bottle

Russian security company Gleg planned to release an upgraded exploit pack for industrial control software that incorporated a raft of new vulnerabilities released by Italian security researcher Luigi Auriemma, who found 30-50 (depending on the report you read...) vulnerabilities in four SCADA products.

<http://seclists.org/bugtraq/2011/Mar/187>

Vulnerabilities in some SCADA server softwares From: Luigi Auriemma <aluigi ()
autistici org>

Date: Mon, 21 Mar 2011 16:16:26 +0000

The following are almost all the vulnerabilities I found for a quick experiment some months ago in certain well known server-side SCADA softwares still vulnerable in this moment. In case someone doesn't know SCADA (like me before the tests): it's just one or more softwares (usually a core, a graphical part and a database) that allow people to monitor and control the various hardware sensors and mechanisms located in industrial environments like nuclear plants, refineries, gas pipelines, airports and other less and more critical fields that go from the energy to the public infrastructures and obviously also the small "normal" industries. In technical terms the SCADA software is just the same as any other software used everyday, so with inputs (in this case they are servers so the input is the TCP/IP network) and vulnerabilities: stack and heap overflows, integer overflows, arbitrary commands execution, format strings, double and arbitrary memory frees, memory corruptions, directory traversals, design problems and various other bugs.

The exploits targeted seven vulnerabilities in SCADA systems made by Siemens, Iconics, 7-Technologies and DATAC.

The issue here is the attention being paid to SCADA vulnerability research rather than the results of that research, at least for the moment.

Bottom line: you don't have to be a specialist to start finding holes.

The image is a screenshot of a presentation slide from ESET. The slide has a dark teal header with the ESET logo on the left and the website address 'www.eset.co.uk' on the right. The main content area is white and features the title 'SCADA Pain Points' in a teal font. Below the title, there is a list of pain points under the heading 'Difficulties in maintaining best practice'. The list includes three bullet points: 'Patching', 'Patch testing', and 'Resetting/rebooting'. Below the list, there are two more lines of text: 'Redundant pathway issues' and 'Siemens and its hardcoded passwords'. The slide has a light grey footer bar.

Critical control systems and any systems not connected to the Internet may be left unpatched. Sometimes it's difficult to test the patch fully (yes, there are still sites that test patches before they're implemented across the site...) and check that it will not hamper operations. Sometimes, it's simply too difficult or expensive or disruptive to take a system off-line. And, of course, it's often assume that systems not connected to the Internet are not subject to outside influence and are therefore don't require protective measures that are considered baseline on other systems.

Sometimes, there's a nasty case of cognitive dissonance where a machine runs software/systems and the contracts with the vendor actually forbid the use of other software.

Future Shock

Hair gap issues in an age of connectivity

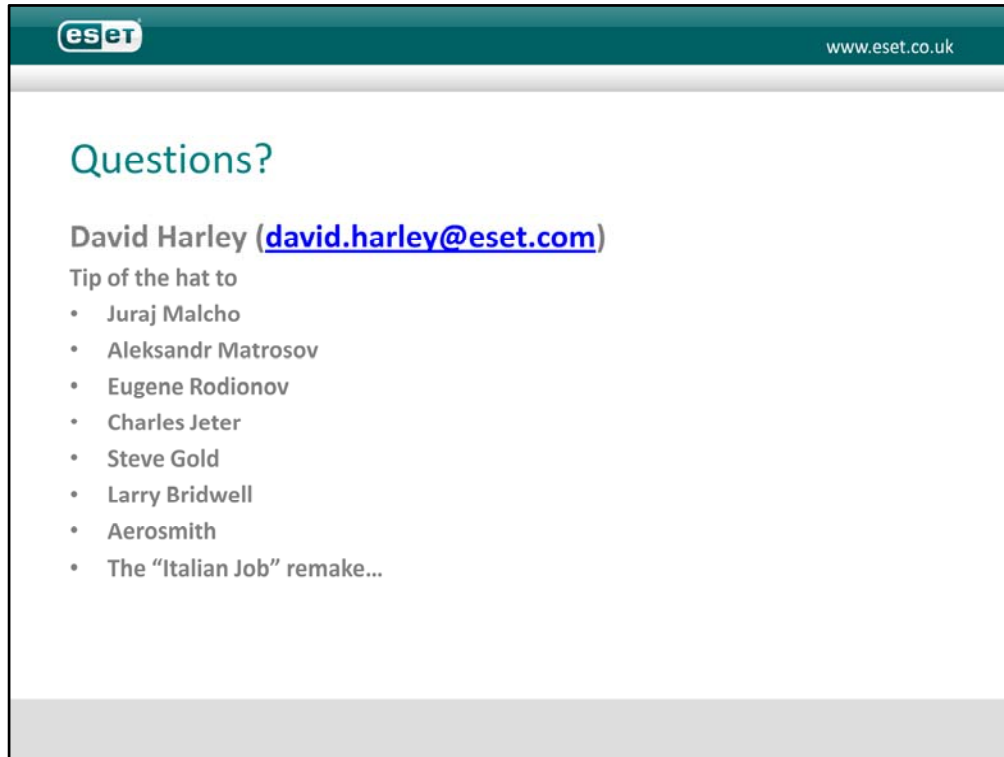


No! Air Gap! Air Gap!

Future Shock

- Air gap issues in an age of connectivity
- Connectivity versus segmented networks
- Referred pressure from connected systems -> backend systems
- Critical data flow /feedback hampered by problems on peripheral "non-critical" systems.
- Has it happened?
- Could it happen?

Everything is connected, directly or indirectly. Issues like power failures and maintenance problems on infected sites generally generate more heat than light in terms of PR exposure. But of course a maliciously engineered disaster is entirely possible. Otherwise, a lot of movie scriptwriters and security prognosticators would be out of work.



The image is a screenshot of a webpage from ESET. At the top left is the ESET logo, and at the top right is the URL www.eset.co.uk. The main heading is "Questions?". Below this, it says "David Harley (david.harley@eset.com)". Underneath, it says "Tip of the hat to" followed by a bulleted list of names: Juraj Malcho, Aleksandr Matrosov, Eugene Rodionov, Charles Jeter, Steve Gold, Larry Bridwell, Aerosmith, and The "Italian Job" remake...

Links:

- http://www.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf
- <http://www.eset.com/us/resources/white-papers/TDL3-Analysis.pdf>
- <http://securityweek.com/shortcuts-insecurity-lnk-exploits>
- http://www.eset.com/us/resources/white-papers/The_Evolution_of_TDL.pdf
- <http://www.securityweek.com/stuxnet-sux-or-stuxnet-success-story>
- <http://blog.eset.com/?s=stuxnet>
- <http://en.wikipedia.org/wiki/F.I.N.E.>