WHY DOES DUQU MATTER?

The short answer is that if your PC got infected by Stuxnet last year, you were just collateral damage, unless you were operating a very specific set of uranium enrichment centrifuges. If you get Duqu this year, your network is under attack from a CIA/Mossad operation. They might seem a little outrageous, but bear with me while we get into the weeds of what Duqu is all about. I will lay out a set of assertions that lead to the conclusion that Duqu really is the "precursor to the next Stuxnet" as Symantec say in their whitepaper.

1. Stuxnet was created by the CIA and the Mossad

Although no one has officially claimed responsiblity for Stuxnet, both the U.S. and Israeli governments have done everything but take offical responsibility. Neither government has ever denied responsibilty, even when directly asked. In fact, officials in both governments have been reported as breaking out in big smiles when the subject comes up.

2. Duqu is from the same team that created Stuxnet.

The first clue that Duqu is from the Stuxnet team is the similarities between the rootkit components in both pieces of malware. The folks who have studied the two most closely are sure that Duqu is based on the Stuxnet component's source code. Despite what you may have read on the internet, the actual source code to Stuxnet is not publicly available. Some folks have reverse-engineered some of the Stuxnet source code from the binaries that are available, for various technical reasons, I'm sure that these don't serve as the basis for Duqu.

Duqu even has a fix for a bug in Stuxnet. Also, the only two pieces of malware in history to install themselves with as Windows device drivers with legitimate, but stolen, digital

certificates are Stuxnet and Duqu. Both Stuxnet and Duqu were active in the wild and managed to evade detection for many months. While that's not unheard of for malware, it is another point of similarity.

Stuxnet targeted a specific industrial control system (ICS) installation (the Siemens PLCs that were used to control the centrifuges at Natanz). Here's the lastest on what Duqu targets:

> Some of the companies affected or targeted by Duqu include the actual equipment that an ICS would control such as motors, pipes, valves and switches. To date, the vendors that make the PLC, controllers and systems/applications found in control centers are not yet affected, although this information could change as more variants are identified and these vendors look more closely at their systems.

There are no other instances of computer malware that target these sorts of installations.

3. Stuxnet was a worm, Duqu is not.

Stuxnet was a very aggressive computer worm. It had to be to jump the "air gap" that protects a secure ICS such as the system that ran the Natanz installation. When Stuxnet was discovered, the A-V vendors quickly discovered millions of computers had been (benignly) infected with Stuxnet. Duqu, on the other hand, has been found on only a handful of computers. Interestingly, no one has yet discovered the dropper, that is, the program used to place the Duqu rootkit on the infected machines. This is almost certainly because Duqu is being placed on these machines via a spear phishing attack. In spear phishing, specific targets are chosen and the attack is customized to the target.

4. Duqu is being used to download a RAT (Remote Access Trojan)

The rootkit component was used to download a standalone program designed to steal information from the computer that it has infected (including screenshots, keystrokes, lists of files on all drives, and names of open windows). Duqu is doing computer network reconnaissance. The information gathered by Duqu is very useful for planning future attacks. Before the command and control server was taken off-line, Symantec observed Duqu downloading three additional files to an infected machine. The first was a module that could be injected into other processes running on the machine to gather some process-specific information as well as the computer's local and system times (including time zone and daylight savings time bias). Another downloaded module was used to extend the normal 36-day limitation on Duqu installations. The last downloaded module was a stripped down version of the standalone RAT, lacking the key logging and file exploration functionality.

5. Put it all together and it adds up to a well-executed, highly targeted covert operation

For the last ten months, Duqu has been quietly stalking a small number of industrial manufacturers. No one even noticed before early September and it wasn't until last week that the nature of the threat was clear to anyone. Duqu is spying on a handful of companies, gathering data that will be used for the design and development of the true Stuxnet 2.0. One thing we don't know is who the target of Stuxnet 2.0 will be. But I have a suspicion. Nothing indicates that the ultimate target (i.e., Iran) of the Stuxnet team has changed. In August of this year, Iran announced that it had activated its first pre-production set of his newer IR-2m and IR-4 centrifuges. These are the successors to the centrifuges that Stuxnet attacked. If you wanted to do these centrifuges what Stuxnet did to the earlier IR-1 centrifuges, you would need a lot of specific data about the safe operating specs of the various components that go into making advanced centrifuges. If you knew or suspected who was

supplying Iran with these components, you might want to gather some data from the internal networks of those suppliers. That's what I think the point of Duqu really is.

DID DUQU FIX THE BUG THAT REVEALED STUXNET?

Duqu isn't Christopher Lee in *Attack of the Clones*, but it is the newest computer malware to hit mainstream consciousness. It's attracting attention mainly because it is based on the same software source code base as the Windows portion of Stuxnet. If you haven't heard about Duqu, check out the Wired article that first alerted me to its existence. If you are interested in the technical details, you need to read the excellent write-up by Symantec (pdf link).

Unfortunately, the twitterverse, blogosphere, and the computer security profession all seem to be caught up in a hype/debunking/speculation cycle that is spreading more heat than light. The primary significance of Duqu is what it tells us about the operation behind Stuxnet and Duqu, i.e. that it is an on-going enterprise conducting computer espionage and sabotage around the world. The fact that it is rather obviously (though not publicly) run by the U.S. intelligence community should concern everyone. I'll put up a more extensive post later (including a timeline!) detailing what the Duqu phase of the Stuxnet operation tells us about the cyberwarfare strategy of the U.S. and how it is endangering the safety and security of the U.S. and the whole industrialized world. But first, I want to remind everyone how Stuxnet was originally discovered:

> … the VirusBlokAda security firm in Minsk, received what seemed to be a relatively mundane email on June 17, 2010. An Iranian firm was complaining that **its computers were behaving strangely, shutting themselves down and then rebooting**. Ulasen and a colleague spent a week examining the machines. Then they found Stuxnet. VirusBlokAda notified other companies in the industry, including Symantec.

This incident became curiouser and curiouser as Symantec, Langner, and others took apart Stuxnet. There wasn't any obvious reason that Stuxnet would have caused that sort of behavior on an infected computer. I even wondered at the time whether or not Stuxnet's cover was blown intentionally since the perpetrators moved quickly to call further attention to themselves. But, thanks to the good work of the Symantec team, we can surmise something quite revealing about the initial discovery of Stuxnet.

The rootkit component of Duqu is quite similar to, but not exactly the same as, the one in Stuxnet. In both cases, if the infected computer gets rebooted while it is infected, the rootkit wants to make sure that it is running before the operating system is fully loaded. That's why this rootkit (both flavors, Stuxnet and Duqu) is packaged as a hardware device driver. Here's a feature of Duqu's driver that wasn't present in Stuxnet (as described by Symantec on page 4 of the pdf linked above):

> The driver then registers a DriverReinitializationRoutine and **calls itself (up to 200 times) until it is able to detect the presence of the HAL.DLL file**. This ensures the system has been initialized to a point where it can begin injecting the main DLL.

The bolded portion is the new functionality that wasn't present in Stuxnet. As a software developer, this detail tells me a lot. The driver is checking to make sure that the hardware abstraction layer (HAL.DLL) of Windows is loaded before it proceeds with the re-infection routine. The HAL is a portion of the Windows OS that really needs to be loaded before device drivers can function properly. Between the time that Stuxnet was deployed and this later version was compiled, the Stuxnet team identified a problem (a race condition) with their software being loaded before the HAL, probably only under the rarest of circumstances. So they modified their program to take this possible condition into account.
As I thought about this, I realized that the likely impact of the Stuxnet device driver being loaded before the HAL was properly initialized would almost certainly be that the machine would continuously crash and reboot. Look again at how Stuxnet was first discovered (remember it was in the wild for at least a full year before it was noticed by any anti-virus vendor):

> … the VirusBlokAda security firm in Minsk, received what seemed to be a relatively mundane email on June 17, 2010. An Iranian firm was complaining that **its computers were behaving strangely, shutting themselves down and then rebooting**. Ulasen and a colleague spent a week examining the machines. Then they found Stuxnet. VirusBlokAda notified other companies in the industry, including Symantec.

By November 3, 2010 (the compile date of the Duqu component), the Stuxnet team had fixed the bug that led to the discovery of Stuxnet last year. And then went almost another full year without being discovered by the anti-virus vendors. It is likely to be a lot harder to reconstruct what the Stuxnet team has been up to this time around, but it is clear that the operation is on-going and we can assume (unless

specific information turns up pointing in a different direction) that the primary target is still the Iranian nuclear program.

STUXNET: THE CURIOUS INCIDENT OF THE SECOND CERTIFICATE

"Is there any point to which you would wish to draw my attention?"

"To the curious incident of the dog in the night-time."

"The dog did nothing in the night-time."

"That was the curious incident," remarked Sherlock Holmes.

Arthur Conan Doyle (Silver Blaze)

[From ew: William Ockham, who knows a whole lot more about coding than I, shared some interesting thoughts with me about the Stuxnet virus. I asked him to share those thoughts it into a post. Thanks to him for doing so!]

The key to unraveling the mystery of Stuxnet is understanding the meaning of a seemingly purposeless act by the attackers behind the malware. Stuxnet was first reported on June 17, 2010 by VirusBlokAda, an anti-virus company in Belarus. On June 24, VirusBlokAda noticed that two of the Stuxnet components, Windows drivers named MrxCls.sys and MrxNet.sys, were signed using the digital signature from a certificate issued to Realtek Semiconductor. VirusBlokAda immediately notified Realtek and on July 16, VeriSign revoked the Realtek certificate. The very next day, a new Stuxnet driver named jmidebs.sys appeared, but this one was signed with a certificate from JMicron Technology. This

new Stuxnet driver had been compiled on July 14. On July 22, five days after the new driver was first reported, VeriSign revoked the JMicron certificate.

The question I want to explore is **why** the attackers rolled out a new version of their driver signed with the second certificate. This is a key question because this is the one action that we know the attackers took deliberately after the malware became public. It's an action that they took at a time when there was a lot of information asymmetry in their favor. They knew exactly what they were up to and the rest of us had no clue. They knew that Stuxnet had been in the wild for more than a year, that it had already achieved its primary goal, and that it wasn't a direct threat to any of the computers it was infecting in July 2010. Rolling out the new driver incurred a substantial cost, and not just in monetary terms. Taking this action gave away a lot of information. Understanding why they released a driver signed with a second certificate will help explain a lot of other curious things in the Stuxnet saga.

It's easy to see why they signed their drivers the first time. Code signing is designed to prove that a piece of software comes from a known entity (using public key infrastructure) and that the software hasn't been altered. A software developer obtains a digital certificate from a "trusted authority". When the software is compiled, the certificate containing the developer's unique private key is used to "sign" the code which attaches a hash to the software. When the code is executed, this hash can be used to verify with great certainty that the code was signed with that particular certificate and hasn't changed since it was signed. Because drivers have very privileged access to the host operating system, the most recent releases of Microsoft Windows (Vista, Win7, Win2008, and Win2008 R2) won't allow the silent installation of unsigned drivers. The Stuxnet attackers put a lot of effort into developing a completely silent infection process. Stuxnet checked which

Windows version it was running on and which anti-virus software (if any) was running and tailored its infection process accordingly. The entire purpose of the Windows components of Stuxnet was to seek out installations of a specific industrial control system and infect that. To achieve that purpose, the Windows components were carefully designed to give infected users no sign that they were under attack.

The revocation of the first certificate by VeriSign didn't change any of that. Windows will happily and silently install drivers with revoked signatures. Believe it or not, there are actually good reasons for Windows to install drivers with revoked signatures. For example, Realtek is an important manufacturer of various components for PCs. If Windows refused to install their drivers after the certificate was withdrawn, there would be a whole lot of unhappy customers.

The release of a Stuxnet driver signed with a new certificate was very curious for several reasons. As Symantec recently reported [link to large pdf], no one has recovered the delivery mechanism (the Trojan dropper, in antivirus lingo) for this driver. We don't actually know how the driver showed up on the two machines (one in Kazakhstan and one in Russia) where it was found on July 17, 2010. This is significant because the driver is compiled into the Trojan dropper as resource. Without a new dropper, there's no way for that version of the virus to have infected additional computers. And there is no evidence that I'm aware of that Stuxnet with the new driver ever spread to any other machines.

The release of the newly signed driver did exactly one thing: Increase publicity about Stuxnet. The inescapable conclusion is that the Stuxnet attackers wanted to make headlines in July 2010. As Holmes says in *Silver Blaze*, "one true inference invariably suggests others". From this one inference, we can begin to understand

the most puzzling parts of the Stuxnet project. Who would publicize their secret cyber attack on an enemy? Why were there clues to the identity of the attackers left in the code? Why did the last version of Stuxnet use multiple 0-day exploits? Why did the attackers only take minimal steps to hide the true nature of the code? The answer to these questions is relatively simple. The Stuxnet project was never intended to stay secret forever. If it had been, there would never have been a new Stuxnet driver in July 2010. That driver helps put all the other pieces in context: the clues left inside the code ("myrtus", "guava", and using May 9, 1979 as a magic value); the aspects of the code that have led various experts to label Stuxnet as amateurish, lame, and low quality; even the leak campaign by the U.S. and Israeli governments to unofficially take credit for Stuxnet. Rather than being mistakes, these were elements of the larger Stuxnet project.

Stuxnet was more than a cyber attack. It was a multi-pronged project. The design of the code supports the overall mission. The mission included a publicity campaign, or as the military and intelligence folks style it, a PSYchological OPeration (PSYOP). Unlike a typical malware attack, Stuxnet had (at least) two distinct phases. Phase 1 required a stealthy cyber attack against the Iranian nuclear program. Phase 2 required that the effects of that cyber attack become widely known while giving the perpetrators plausible deniability. That may seem a little strange at first, but if you put yourself in the shoes of the attackers, the strategy is more than plausible.

In fact, the attackers have explained it all. Take a look back at the story told in the New York Times article on January 15, 2011. According to the NYT, the Stuxnet project started as an alternative to an Israeli airstrike:

> Two years ago, when Israel still thought its only solution was a military one and

> approached Mr. Bush for the bunker-busting bombs and other equipment it believed it would need for an air attack, its officials told the White House that such a strike would set back Iran's programs by roughly three years. Its request was turned down.

Couple that statement with the reason the article appeared when it did:

> In recent days, American officials who spoke on the condition of anonymity have said in interviews that they believe Iran's setbacks have been underreported.

Imagine that you're an American policymaker who has to choose between launching a cyber attack and allowing a close ally to launch an actual military attack. If you choose the cyber attack option, how will anyone know that you've succeeded? If no one knows that you've successfully delayed the Iranian nuclear program, you'll be vulnerable to right-wing attacks for not doing enough to stop Iran and the pressure to bomb-bomb-bomb of Iran will grow. There's another reason to publicize the attack. If you're a superpower who starts a cyber war, you have to realize that your country contains a lot of very soft targets. You would want to make a big splash with this malware so that your industrial base starts to take the cyber war seriously. So, from the very beginning, the project included planning for the inevitable discovery and understanding of the Stuxnet malware. Just like the spread of the malware itself, the psyop will be impossible to directly control, but easy enough to steer in the appropriate direction. The attackers likely didn't know it would be Symantec and Ralph Langner who would start to unravel the exact nature of the Stuxnet malware, but they knew someone would. And they knew they would be able to get the New York Times to print the story they wanted to get out (I'm not demeaning the work of the reporters on this story, but I would

hope they realize that there is a reason they aren't being investigated for publishing a story about our efforts to undermine Iran's nuclear program and James Risen was).

SARAH PALIN: GIBBERISH WE CAN BELIEVE IN?

Energy is supposed to be Sarah Palin's strong point, right? After all, she is the Governor of Alaska, and more to the point, was the chair of the Alaska Oil and Gas Conservation Commission, the agency that is supposed to "protect the public interest in exploration and development of oil and gas resources, while ensuring conservation practices, enhancing resource recovery, and protecting the health, safety, environment, and property rights of Alaskans." But when she was asked about ensuring that the fruits of domestic oil drilling would go to the domestic market, her answer was complete gibberish.

THE STRANGE CASE OF HIWA ABDUL RAHMAN RASHUL (PART 2)

In part 1, I laid out the facts surrounding the detention and illegal transfer of Hiwa Abdul Rahman Rashul. In this post, I want to demonstrate why this case matters. There is a pattern to the Bush/Cheney Administration's illegal usurpation of executive power. Because the pattern broke down in this case, the

strategy behind that power grab is laid bare. The struggle within the administration over the disposition of Rashul and the way it was resolved helps to illuminate the true nature of the current regime. Perhaps it leaves an opening to unravel the authoritarian infrastructure that has been built within our country in the last eight years.

THE STRANGE CASE OF HIWA ABDUL RAHMAN RASHUL (PART 1)

In June 2004, Hiwa Abdul Rahman Rashul had his 15 minutes of fame when Secretary of Defense Donald Rumsfeld answered questions at a press conference about the detainee known to American soldiers only as Triple X, the first ghost detainee transferred from CIA custody to the U.S. military. Rashul was suspected of being a member of Ansar al-Islam, a violent Kurdish Sunni Islamist movement opposed to the dominant Kurdish groups of northeastern Iraq. The real story of Hiwa Abdul Rahman Rashul wasn’t his terrorist past or his time as a ghost detainee of the DOD, but his treatment by the CIA in between.