This is my unofficial English translation of our 2nd article on supply chain security. Rolv R Hauge, Eirik Dragvik and myself (Tor Houghton) wrote it for Telenor Norge's Digital Sikkerhet 2021 (this translation, version 1.00). All copyright original rights holder. As before, the double connotation of the Norwegian word "nettene" does not translate well, and any use of "company" should be read as "organisation" and vice versa. Some additional footnotes have been added to this edition.

Supply chains – a known risk that requires conscious decisions

When Nights Are Getting Longer Part 2 – a subject matter deep dive: Is supply chain risk something we actively consider and take a position on and is it part of the company's risk discussions? Are discussions on digital security and supply chains so challenging that managers and decision makers are unable to partake? In When Nights Are Getting Longer Part 2, we will show that technology and an understanding of how it can be used are important for making conscious choices. The past year has offered plenty of examples of supply chain attacks.

Solar Storm

When we wrote the chapter *When Nights Are Getting Longer* in Digital Sikkerhet 2020, SolarWinds Inc.'s systems had already been compromised by a threat actor – later attributed to Russia – for a year and a half. The threat actor had used SolarWinds' mechanisms for software development and distribution of updates to get their malware delivered to thousands of customers (March 26, 2020).

As recently as December 2020 it was still possible to download the compromised software from SolarWinds Inc. The certificate that was used to sign it had still not been revoked. It is easy to see that the handling of the incident – at a time when SolarWinds knew what had happened – was not optimal. As outsiders and in hindsight, it is also easy to criticise them for it. Just as interesting is the fact that only one of the 67 antivirus engines on Virus Total[1] was able to detect the malware at this time.

Did SolarWinds understand their responsibilities and the risk exposure that they represented to their customers? Should they have? Many of SolarWinds Inc.'s customers are Managed Service Providers (MSPs) who in turn service hundreds or thousands of customers with this software. Did SolarWinds Inc. have the necessary expertise and capacity to manage this risk? And did they have sufficient control over what was added to their code base?

There is reason to believe that robust security checks in the build environment could have given an indication that something was wrong. Good practice in the area is well established – such as

[1] https://virustotal.com

various forms of automated code analysis and vulnerability scanning – and could probably have helped to avert or limit the impact of the attack. For changes in particularly sensitive components of the software or where changes are made to a significant portion of the code base, human peer review, with or without tool-based detection, may be a necessary insurance.

Just an innocent victim?

There is also reason to question if the largest customers of SolarWinds should have exercised a greater degree of control over the deliverables, or required that thorough security quality controls be carried out by a qualified third party before shipping the product. MSPs, by their one-to-many position in the value chain and with privileged access to their customers, are an effective and well-known attack vector[2].

To uncover a successful supply chain attack in received software, however, resource- and competence-intensive tools must be used if the supplier's development or build environment is compromised and the compromised software appears completely authentic. With the significant amount of software tools an average business actually uses, many will hesitate to perform such in-depth verification.

Many will only use basic integrity checking and malware scanning when receiving software. It is a start, but for crucial software deliverables, it is becoming increasingly clear that further efforts are required to meet a risk that manifests itself as increasingly serious incidents. However, more advanced measures are not just about the ability to detect compromised software and prevent it from being used – good security monitoring can also contribute to early detection. This can be of great value, as it often takes some time from initial compromise through the supply chain, until the threat actor uses the established footholds against their target selection.

The past year

The attack on SolarWinds Inc. received a great deal of attention, but was just one of several serious successful supply chain attacks in the past year. We note, among other things:

Discovered	Event	Entry	Affected Software	Type of Operation	Synopsis
May 2021	The website of the president of Myanmar	Public website	Font definition package	Watering hole attack	Attackers used unknown vulnerabilities to modify a popular font package published on the presidency's website.
May 2021	Fujitsu ProjectWEB	Unknown	Storage and collaboration platform (SaaS)	Unknown	This type of storage and collaboration is widely used by Japanese authorities.
Apr 2021	CodeCov	Misconfigured Docker image build	CodeCov Bash Uploader script	Theft of keys from the environment	The tool is used for security and integrity

[2] https://baesystemsai.blogspot.com/2017/04/apt10-operation-cloud-hopper_3.html

		process		the tool is used in	testing of code in the development environment of over 29 000 customers. Enabled, among other things, the theft of keys and other authentication information from the users of CodeCov Bash Uploader, which allowed further access to these customers' own development environments and products.
Mar 2021	SITA	Unknown	Passenger management system and database for several airlines	Theft of personal information	SITA provides passenger management services to approximately 90% of the world's airlines, and had passenger information dating back at least 10 years. For some this included passport and credit card information, but for others "only" basic personal information[3].
Mar 2021	Google Play Store	Upload of malware-infected mobile applications	A handful of mobile applications	Theft of bank accounts and circumvention of 2-factor authentication	Unfortunately, this is not the first example of mobile apps with deliberately encoded malware succeeding in being published in reputable app stores.
Feb 2021	Birsan research	Open source libraries	Several	Proof of concept	A researcher showed that many large companies' software build pipelines could be tricked into retrieving external content, if a publicly available package with the same name as the internal (legitimate) packages was published.
Des 2020	VGCA (SignSight)	Public website	Digital Signature Toolkit	Targeted at authorities and selected companies	Digital signing software, published on the website of the Vietnamese authorities' official digital certificates issuer, was compromised with a back door. This gave access to the many Vietnamese companies that downloaded and used the software in question, and further attacks against these.
Nov 2020	VeraPort	Compromised website (watering hole attack)	Browser plugin	Targeted at authorities and selected companies	Using stolen certificates for code signing and exploitation of vulnerable "VeraPort" web services,

[3] Recommended reading with relevant description and technical analysis: https://blog.group-ib.com/colunmtk_apt41

					the threat actors managed to spread a compromised version of client software that is widely used in South Korea for interaction with banks and other entities.
Jul 2020	Twilio SDK	Misconfigured storage service in public cloud	Cloud-based communication (CPaaS)	Information theft (communication)	A misconfigured "S3 bucket" belonging to Twilio enabled the threat actor to upload a modified version of a library for programmatic interaction with Twilio TaskRouter. The attack had limited consequences, but illustrates the importance of having correct security configuration of the cloud services that one consumes and that whoever uses such services in the cloud is responsible for this.
Jun 2020	Goldenspy	Software with back door	Business application	Targeted at selected western companies	The threat actor compromised Western companies in China by distributing malware-infected tax software.

Turtles all the way down

Many client applications are now browser-based. This has a natural explanation. Most browsers are now considered full-fledged operating systems with important security mechanisms and not least standardised programming interfaces that, regardless of whether the browser runs on Windows, Mac, Linux or a mobile phone, can run the same application code. The costs of developing such applications are thus greatly reduced.

A contributing factor, which helps reduce development time and cost, is the dynamic use of third-party libraries such as Node, JQuery and Chartbeat. By *dynamic,* we mean that libraries are either downloaded at application build time or when it is running in a browser. On the one hand, such libraries often contribute to more secure code, as they often help programmers to “do things right”, while on the other they increase the risk of being compromised through the supply chain. This was duly demonstrated in February when a security researcher described how he had exploited this type of dynamic download by publishing packages of conceptual “malware” to different public frameworks (NPM, RubyGems and PyPI) with the same, or almost the same, name as those that several larger technology companies themselves used for their internal modules[4].

[4] https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610

For a small business, without its own capacity for in-depth quality control of third-party components, or the finances to purchase such assistance, dynamically retrieving the latest published version from public sources may have a greater security benefit than downside. At least one is sure to get the latest security improvements, despite the fact that one also runs the risk of getting the latest supply chain attack as “corpse in the cargo”. For other companies, the risk – both their own and customers’ – may imply a greater degree of control.

This justifies the use of resources for in-depth quality control that can reduce the likelihood of a hitherto unknown supply chain attack slipping through.

Dynamic retrieval of third-party components requires a risk-based balance between trust and control. This must be based on an understanding of one's own and customers’ risk tolerance. Telenor Norway has first-hand experience of how this can play out, as a software delivery from a supplier turned out to contain malicious code from a library that was dynamically retrieved at build time, several levels down in the dependency hierarchy. For our part, it was fortunately discovered before the code came into our service production. In this case it could do no harm either, as it was written for a completely different and specific target in mind. However, we note the threat actor's financially motivated willingness to place code, that ends up in thousands of software builds, and “burn” his own small open source code project, only to reach one software that used this library.

Open source – supply chains include more than classic “suppliers”

Another – and quite different – supply chain attack that attracted some attention in early 2021 was the sneaking of “vulnerabilities” in the Linux kernel code by students at the University of Minnesota[5]. The students took as their starting point the good reputation of the university among Linux kernel developers, and provided code entries with deliberate “vulnerabilities” which were then accepted by authorised kernel developers and included in the official Linux kernel.

The incident sparked a major debate about whether it was ethically correct to conduct such a security experiment. In relation to supply chains, the case illustrates how vulnerable many basic open source projects, on which much modern IT is based, really are. The fact that *everyone* can see the source code does not necessarily provide good security control, although in this case it was eventually discovered. The degree of control, the effort for security improvements and the ability to address vulnerabilities, depends on available resources in the projects.

There are surprisingly many open source projects of an almost global critical nature that depend on the unpaid efforts of a few enthusiasts. For example, the world discovered in connection with Heartbleed[6] that the popular SSL library *OpenSSL* was in fact maintained by only one person. Fortunately, in recent years, several commercial companies have become aware of the need to

[5] The Verge: How a University Got Itself Banned from the Linux Kernel – https://www.theverge.com/2021/4/30/22410164/linux-kernel-university-of-minnesota-banned-open-source
[6] https://heartbleed.com/

support these projects – and that giving something back when producing highly profitable products with FOSS (Free and Open Source Software) components is only reasonable.

Do you know what you are getting? (Software Bill of Materials)

It is becoming increasingly common in agreements to not only require that the physical deliverables are specified in detail in the *Bill of Materials*, but also that the software and what it contains is described in a *Software Bill of Materials* (SBOM). An SBOM describes all software components included in the deliverable, including sub- and sub-components with precise specifications such as version and build numbers. There are many benefits to knowing what you are actually getting delivered at this level. The most immediate can be linked to security quality assessment of a deliverable both before and after the purchase decision has been made, *input/supplementary data to asset inventory*, vulnerability management and a basis for security verification of the received item. Shouldn't it be completely obvious to document what the deliverable contains?

Software deliveries can often have a comprehensive component list and hierarchy. It is therefore necessary, not least to be able to make efficient use of the information, to produce the SBOM as a machine-readable structured data set. Several standard formats and tools for reading, writing and converting between them are openly available. We believe that most companies that acquire software should require such a "content declaration". The US Department of Commerce has a good introduction to SBOM, with many good references[7].

The choices we make

It is now ten years since the first really big supply chain attack we know of; the attack on RSA, where Chinese state actors stole so-called seed values for SecurID tokens. With these, threat actors were able to get past the two-factor authentication of all SecurID customers, including a subcontractor to Lockheed Martin, Level 3 Communications and Northrop Grumman[8]. Following this, Lockheed Martin established the concept of the Cyber Kill Chain[9] which provides a structured review of seven generic main steps in cyber operations. Later that year, Lockheed Martin chose to establish and publish a model for a modern defensible security architecture[10].

The sustained flow of successful supply chain attacks – from the RSA attack to the present – underscores one of our key messages from *When the Nights Get Long* in Digital Sikkerhet

---

[7] U.S. Dpt. of Commerce – National Telecommunications and Information Administration: https://www.ntia.gov/SBOM

[8] This year, former employees of RSA have finally been able to talk about the attack, and we have gained more knowledge about it https://www.wired.com/story/the-full-story-of-the-stunning-rsa-hack-can-finally-be-told/

[9] Lockheed Martin – The Cyber Kill Chain: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[10] Lockheed Martin: Defendable Architectures https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Defendable-Architectures.pdf

2020: As a responsible supplier, you *must* understand the risk your deliverables pose to your customers' values. Without this, you also have no basis for understanding your own attractiveness to threat actors, or your own protection requirements.

A certain supply chain risk does not escape any business, but the degree of exposure and risk is the result of conscious choices. Which suppliers do you choose, and on what basis? What requirements do you make and how? How much do you invest in control activities? How do you prepare for the next *supply chain attack* when it hits *your business*?

What happens when the supplier is sold?

Businesses are bought and sold. In 2021, the sale of Bergen Engines received a great deal of attention in the Norwegian media. This was based on concerns about the buyer's intentions in relation to Norwegian national security interests. However, the problem is not limited to national security.

The sale of a supplier, in whole or in part, can significantly affect the supply chain risk you are exposed to and in several ways. The way the supplier business is run can change significantly, and affect both delivery capacity and commercial conditions. Where the company's various functions are located, or where the headquarters of the acquiring company is located, can affect both the security of the deliverables and how your information is handled by the supplier. Both jurisdiction, corruption index and societal stability come into play here. Objective conditions in the supplier business can also be changed by, or as a result of, a business transfer; for example, the supplier's financial soundness and objective security. All in all, this is of great importance to you as a recipient of the supplier's products or services.

In addition to ordinary supplier assessments and price competition, long-term stability and the probability that the supplier business will be transferred to any third party should be taken into account in the choice of supplier. Especially where these conditions can be of great importance to your business. Such an assessment requires, among other things, insight into the supplier company/supplier industry group's composition, their views on their own core business, the company's foreseeable and future capital needs. If the product or service you buy, or the unit that produces it, is on the side of the supplier's core business, there is an increased risk of a change of hands. Studying trends and tendencies towards consolidation in the supplier's market and industry can also provide important indications.

No matter how thorough the preparation is when entering into an agreement, future changes of ownership, consolidations and business transfers are difficult to predict. If it does occur, however, a risk assessment should be made as to whether the supply chain exposure and risk have changed significantly. For a period, or as a permanent measure, it may be necessary to manage the supplier relationship less trust-based to one with a greater degree of actual control and verification. For example, through strengthened supervision of the supplier company's security level and the security quality of the deliverables. If the identified risk borders on an

unacceptable level, or the supplier opposes supervision that you have deemed necessary, changing supplier will be the last resort. This is rarely desirable, but cannot be ruled out.

Where an increased risk of changes in ownership has been identified with important suppliers, it should be considered whether risk-reducing measures should be implemented before the possible change of ownership occurs. When defining supplier importance, you must take into account the supplier's and the deliverable's negative impact potential on your business, your services and your customers. Not only how critical the supplier's contribution is to maintaining your own production. If such risk is identifiable at the conclusion of the contract, there may be room to maneuver in drafting of the contract, for example by pre-emptive rights to the relevant part of the supplier. Insourcing that shortens the supply chain can in any case be considered a risk-reducing measure if a supplier business, or part of it, is offered for sale. A more pragmatic and far more common approach is to have standby agreements with alternative suppliers. The price for this will often be a certain minimum purchase from a less affordable standby supplier. A permanent setup with two or more suppliers can also be a risk-reducing measure. The real ability to fall back on just one of them, and not least the increase in risk by exposing oneself to several supply chains, must be weighed against the safety gain of such a measure.

Myth 1 - "We can not do anything about APT attacks"

Pointing out that it was "a very advanced attack" has become a common explanation by compromised companies. There is, of course, a hint of truth in that, although the explanation is diligently used in the media even when neither the actor nor the technique was particularly advanced. It is true that it is very difficult to protect oneself against the really advanced and determined threat actors. However, it is possible to reduce the likelihood of the attack succeeding, persisting, spreading, or having catastrophic consequences. A responsible business can not put its hands in the air and give up "because APT".

1. **Make it expensive** for them, by maintaining basic safety hygiene. Security patching and secure configuration force the attacker to use more competent resources. APT actors have large resources at their disposal, but they must also prioritise. It costs them much more to find and use hitherto unknown vulnerabilities. Although it is legitimate to take the risk and exposure of the vulnerabilities into account when prioritising patching, it is worth noting that it is now quite common to combine vulnerabilities with "low" risk scores, so that together they enable a serious security breach. The patching of *low* and *medium* vulnerabilities can therefore not be postponed indefinitely[11].

2. **Delay** them. Segmentation and minimised exposure both externally and internally, forces the attacker to spend time moving through the infrastructure from the first point of attack. Among other things, this gives you more time to observe and limit an attack during development. Know, and be aware of, what you expose to whom and how – all exposure should be based on real requirements.

[11] Chris Gates – "Pentesting from LOW to PWNED" https://www.youtube.com/watch?v=u68QvWXYW_Q

3. **Know your infrastructure**. A world-renowned APT[12] has itself pointed out that they have usually become better acquainted with the target's infrastructure than the target's own IT department.

4. **Instrument for detection**. Robust security monitoring requires visibility and analysis both on log events, traffic (also "internally") and on *hosts/workloads*.

5. **Think layered security** in design, architecture and threat modeling. It should never be enough for one mechanism to fail for an attack to succeed. Remove cross-system dependencies where possible so that the scope and effect of a successful attack is limited.

6. **Plan for continuity**. *Offline backup* and concrete recovery plans can help reduce the business damage effects. Although many APT attacks target retrieval and information theft, we have seen several examples of APT tools getting into the hands of organised criminals – where destructive ransomware attacks are highly relevant.

Myth 2 - "We have to fix the users!"

We constantly hear that "users are the weakest link". If we look at the input vectors for cyber attacks, there is no doubt that humans and humanity are being exploited. *Drive-by* attacks when surfing, emails with infected attachments and links that direct the user to both malware and phishing, are vectors that are often attributed to the fact that it is the human that has failed. The conclusion is thus often that we must "fix the person" to prevent these things from happening. Several *awareness campaigns*, phishing tests and similar activities are intended to enable users to better act with healthy skepticism and critical sense in their encounters with attackers. It's not stupid, but these activities do not take us over the finish line. We only continue to play in the arms race where the attackers become increasingly creative and cunning, and still manage to fool the users.

"Do not open attachments from people you do not know" and "do not click on links" are well-meant advice, but for many this is precisely part of everyday work and way of work. "If in doubt, feel free to call the sender to verify," we "security experts" have also said. The advice requires that you are in fact in doubt about the authenticity of the inquiry, and that you take the time to check this in a hectic everyday life. In addition, *zero click* vulnerabilities appear from time to time that can be exploited. Two of these were found in Apple's iOS in May[13]. In such cases, the user has little to contend with.

---

[12] Rob Joyce, Head of NSA Tailored Access Operations: https://www.usenix.org/conference/enigma2016/conference-program/presentation/joyce

[13] In fact, Citizen Lab write that these were actively exploited since February: https://citizenlab.ca/2021/09/forcedentry-nso-group-imessage-zero-click-exploit-captured-in-the-wild/

We believe that the vast majority of companies must go further in using technological tools that take into account that the use of online services and emails with attachments are part of everyday work and work processes. Most employees are not hired to spend all their time and energy on avoiding being fooled, but have other tasks. The threat actor has only one task to contend with: How can I manage to fool the recipient? The threat actors also have a scale advantage – a few attackers can point the same weapon at many victims.

Established tools such as e-mail scanning and SSL inspection can look for known malware and malware patterns. However, greater emphasis must be placed on preventing the consequences from becoming fatal when something slips through. We like to refer to this “final fortification” as *containment*; measures you take to prevent the harmful *effect* from occurring, or at least to limit it to an isolated environment.

Authentic8 Silo and *Cloudflare Browser Isolation* are examples of the *Remote Browser Isolation* (RBI) service category. Here, a virtual “disposable” browser will run in their (cloud) environment, and any exploitation of browser vulnerabilities or other programs that a download through the browser may interact with, remains isolated on their platform and is gone next time the user starts a new browser session. *HP Sure Click* (formerly *Bromium*) is an example of process isolation (locally) on the user’s PC where specific processes, such as browsers, document readers and Office applications, can be controlled to run in a separate micro-kernel. This limits the damage and most likely prevents the damage from unfolding.

Even very simple mechanisms can have a good effect; such as:

- Associate all script file types, such as “.bat” and “.ps1” in Windows (if they were allowed into the user’s PC platform) to open in a text editor;
- Remove users’ administrator privileges;
- Harden the client;
- Keep clients up to date; and
- Use a secure DNS service that filters known malicious domains, such as *OpenDNS Umbrella*, *Telenor Nettvern*, *Cloudflare* 1.1.1.2 or *Quad9* 9.9.9.9.

Security awareness training still has its place in the security program – there are many other ways to be fooled.

<end>