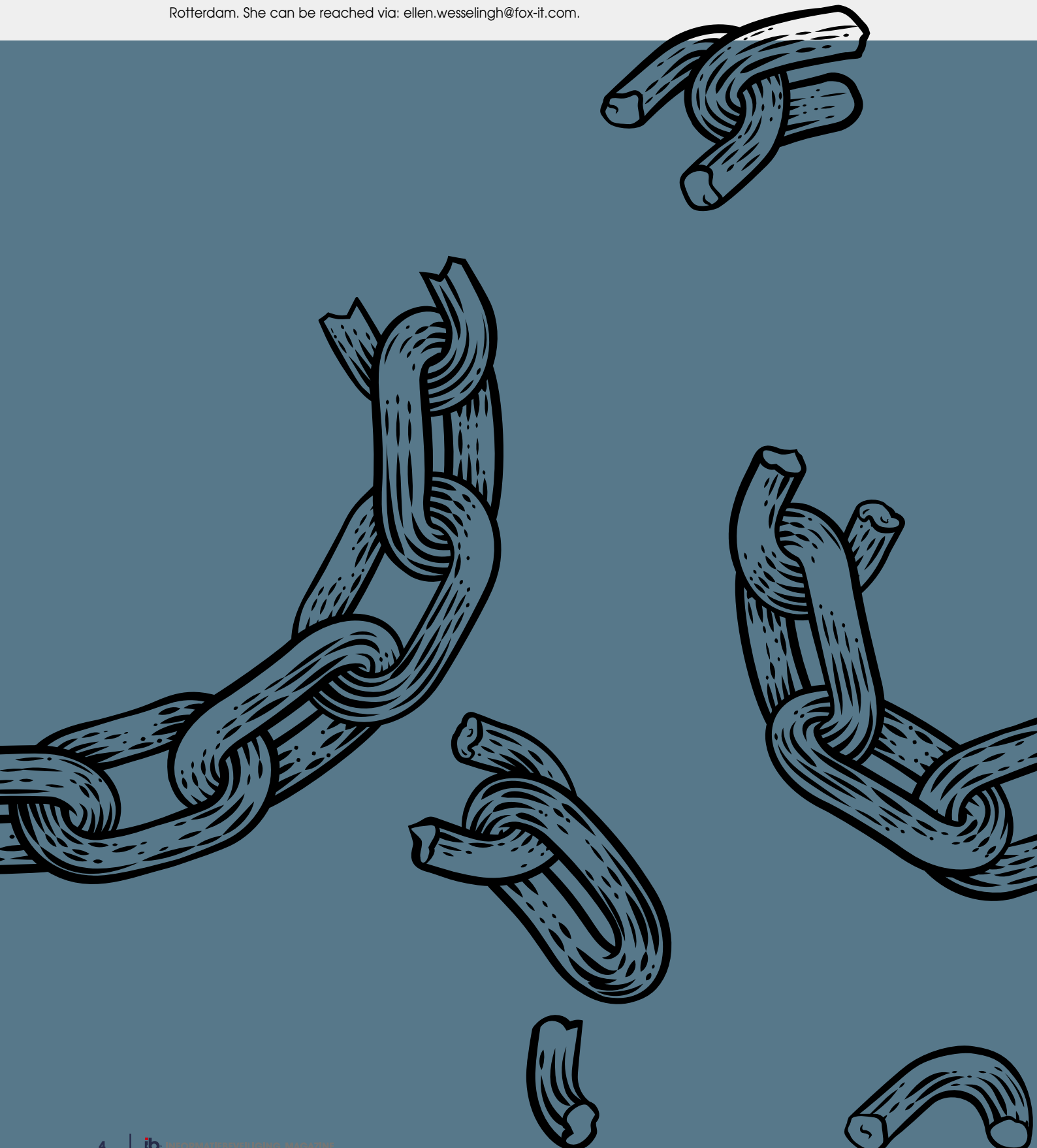
Author: Ellen Wesselingh is senior with subject matter expertise in product security certification, Member Workfield Committee HBO-ICT bachelor NSE program in Delft and Member Workfield Committee HBO-ICT bachelor Computer Science program in Rotterdam. She can be reached via: ellen.wesselingh@fox-it.com.



Managing Supply Chain Cybersecurity Risk through Life Cycle Modelling

Eliminating the weakest link

In recent years, supply chain issues for products with digital elements have increased, which poses a problem for the assurance of the Integrity of these products and the confidelity of the data contained in these products. To illustrate the problem, this article provides a number of examples of supply chain attacks that have happened over the last couple of years. The article also provides a model for supply chain risk analysis. This model is based on an existing model from 2013, which is amended with a level of abstraction to ensure the model is as complete as possible. A risk analysis that has been conducted with the support of this model, complemented with an additional analysis of the remaining risk, should provide a sufficient argument that the supply chain is secure enough for its purpose.

This article discusses recent developments in products with digital elements that may lead to security issues when systems with these elements are deployed. It also discusses possible solutions. One problem is not addressed: misinformation. While a major problem, misinformation makes the product less reliable in the eyes of the beholder. That means it is not the target of this model, because it does not address the security of the system. Misinformation however, may lead users of the system to less secure choices.

It is recommended to use (internationally recognised) standards for interoperability, industry standard development tools for quality assurance, and to re-use architecture, designs, firmware, and software for cost efficiency and development time reasons. However, since the (re-)use of these components expands the supply chain to a great extent, this leads to a higher possibility of supply chain attacks.

In the past ten years, supply chain problems have become more apparent. This can be seen from examples such as Meltdown and Spectre, which have shown hardware architecture choices can introduce the possibility of sophisticated hardware attacks. Kaseya and Solarwinds have shown a similar introduction of attack possibilities through supporting services. Another seminal example is the Dutch case of Diginotar, in which a certificate issuing service was hacked. These examples have made it clear that hackers can attack potentially interesting targets through their supply chains. This is especially true for high assurance products that are potentially interesting targets for Advanced Persistent Threats (APTs), also known as state actors. Such targets with much exposure include industries such as the energy, the financial and the military sector.

Hardware supply chain attacks can be distinguished into two categories. The direct hardware supply chain attacks are executed by actively inserting backdoors and/or trojans in hardware, while the indirect hardware supply chain attacks, such as Meltdown and Spectre, result from genuine design decisions with adverse security consequences. Similar considerations have to be made for firmware and software, with the distinction that

Supply chain attacks are a real risk, especially for high assurance products

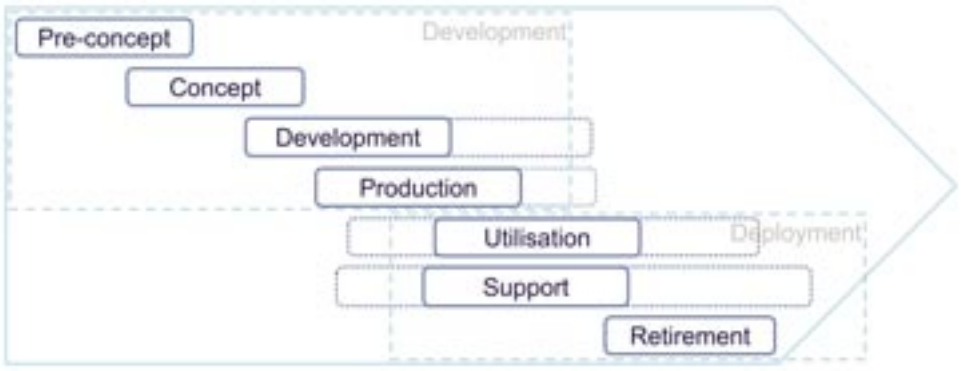


Figure 1: Life cycle model.

these may be updated on production systems, whereas that is usually not possible for pure hardware systems such as IC and ASIC.

A target can be the supply chain for direct organisational operations, such as the production supply chain. But indirect supply chains can also be targeted, affecting services that an organization uses for marketing research. Other similarly indirect operations, such as a financial backend, or a Human Resources Management (HRM) system, may be just as vulnerable. The bottom line is that organisations should be aware of all their supply chains and the way these elements may interfere with critical business operations, even when they are perceived remote elements of the complete operation with all its supply chains.

The primary focus of this article is to develop a supply chain attack framework that addresses the primary process in which a product with IT components is produced. Such products may consist of hardware, firmware and software, a combination of these three or (development) system information or other product data. Supply chain issues that affect the systems which support the development environment are also addressed.

Secondary services such as the supply chain of other office processes or communication processes are not in scope, but may be addressed in future work. The same holds for supply chains of services that are provided to customers.

Methodology used

The research was inspired by a presentation from Andrew Huang for a Blue Hat conference. In his presentation, Huang mentioned a number of attacks for which the articles describing those attacks were analysed for further ideas and references (snowball method). Subsequently, a search for (hardware) supply chain attacks with more generic keywords was conducted to find other supply chain hardware attacks. These attacks were then analysed. No specific search for software issues was done in this phase because of the overwhelming amount of software security problems.

Once a list of relevant hardware supply chain attacks was compiled, a search was conducted to find existing life cycle frameworks and supply chain frameworks. This search yielded a list with possibly relevant frameworks from credible institutions such as ISO, MITRE, NATO, NIST. Once the list was compiled, the most relevant existing framework was chosen. The ISO, NATO and

NIST frameworks mainly address single organisations, and tell how to fix problems in a standardised way. The MITRE framework shows what can go wrong and what the possible solutions can be.

The existing framework used a different life cycle model and was therefore transformed to the life cycle model in use at this company. The chosen framework provided a list of attacks in the various life cycle stages and a list of countermeasures that may be taken against the attacks. The new model was amended with an attack in the retirement stage, the new life cycle model is shown in figure 1.

The model was then validated for usability by applying it to an actual supply chain situation from the stakeholder. Following the application, several new countermeasures were added, and the model was introduced to colleagues. It has since then been used on actual supply chain situations multiple times.

This article describes how the model was re-worked by rearranging the attacks and countermeasures. The framework was developed in several stages. In each stage, the model was validated by peer review with expert colleagues internally. In between those stages, the framework was applied to actual project questions, leading to validation of the model.

In the course of writing this article, information on supply chain attacks and vulnerabilities for firmware and software was sought. One of the examples that was found is Log4J, an Open Source Software (OSS) component widely used in many IT systems, of which the administrators and users were often unaware that it was used in their applications.

Further, an overarching layer of abstraction was added to show more convincingly model completeness. Finally, the model was validated by giving presentations to peers in the field, leading to useful feedback.

Updated model

Figure 2 shows the highest abstraction layer of the model. The model contains different layers of abstraction of the product. The development environment is modelled as relevant for all stages of development: standard, architecture, hardware, firmware, software. Secondary services such as financial backend or HRM system are out of scope for this model.

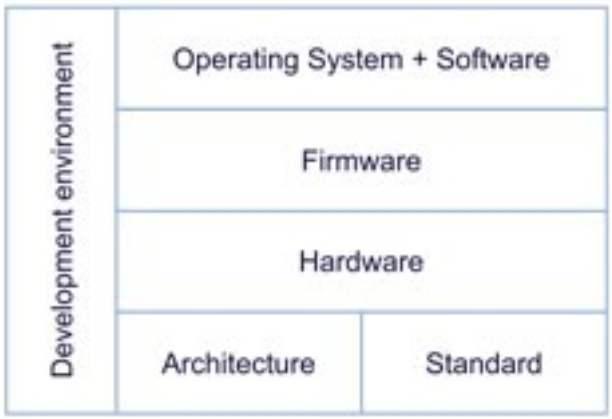


Figure 2: Attack surface categories.

Figure 2: Attack surface categories defines the attack categories in which various attack types can be identified. They also provide specific examples of such attack types. Below is a list of attack types and examples in each category:

- Architecture: attacks as a result of architectural design choices. Examples are Meltdown and Spectre, which may be categorised as either Architecture or Hardware.
- Standard: attacks as a result of vulnerabilities in the standard that is used. An example is Terrestrial Trunked Radio (TETRA). Parts of the TETRA standardised protocol contain vulnerabilities due to government restrictions on the cryptography used.
- Hardware: attacks based on genuine design choices like Meltdown and Spectre. Other examples include attacks based on malicious additions to the design in pre-concept, concept, development and production stages.
- Firmware: attacks based on genuine design issues, such as the Joint Test Action Group (JTAG) interface. This interface is necessary for testing during development and production and to provide updates during support. Other examples are attacks based on malicious additions to firmware design in concept, development, production or utilization/support stages of which Stuxnet is an example. Note that the firmware definition used is the following: makes a generic component a fixed function device. This fixed function device can still be updatable.

- Software: attacks based on genuine design issues such as Log4J, an OSS Java-based logging utility often used in web applications, which had multiple vulnerabilities due to programming issues. Other examples are attacks based on malicious additions to the design in pre-concept, concept, development, production and utilization/support stages, such as typosquatting or dependency confusion. In this article, software consists of operating system and any application running on that operating system.
- Development environment: attacks on data and/or systems, based on genuine design issues in the supplied systems like Kaseya and Solarwinds. Other examples are attacks based on malicious additions in the supplied systems (backdoors, ransomware, trojans, viruses). These issues may also arise in the production environment. On 29 March 2024, a new attack vector in the development environment emerged: the human factor.

The categories, including attack types, are combined with the chosen life cycle model. This life cycle model has seven stages, described below. It is important to note that the stages are not linear.

1. Pre-concept, in which generic (market) research is performed to find the customer needs, requirements and wishes.
2. Concept, in which a Proof Of Concept (POC) is developed to validate the results of the pre-concept research.
3. Development, in which the POC is developed to a Technological Readiness Level (TRL) for production.
4. Production, in which the developed system is produced and delivered to the customer.
5. Utilisation, in which the system is deployed, this stage incorporates the acceptance and installation.
6. Support, in which the deployed system is being maintained in an operational and secure state.
7. Retirement, in which the deployed system is securely destroyed to prevent persistent data leakage.

Note: the development and production stages run parallel in part, and the stages utilisation and support stages run largely parallel. In the definition of the Common Criteria standard, the production stage is considered part of the family Development Security (ALC_DVS).

For each attack type as listed, they are projected on the life cycle stages. Subsequently, the corresponding countermeasures for each attack are added to the model. All countermeasures are categorised by their applicability to the various attack types.

Combined, the attacks and countermeasures provide a model which can be used for supply chain attack risk analysis. This method is a qualitative method, which means that it does not calculate the risk that an attack is feasible. However, it shows any remaining risk in the supply chain that can either be accepted or mitigated with measures such as insurance. A simplified model is presented in Figure 3: Life cycle with attack type categories. This figure assumes an ideal world in which hardware is developed and produced first. The full model with the details of attack types and countermeasures can be found in the appendix.

The model was further elaborated in multiple steps:

1. Reshuffled the original list of attacks into the newly defined categories Architecture, Standard, Hardware, Firmware, Software, Development environment. Discussed the attack list with expert colleagues. Analysed each attack for essentials such as the entity that is in control when the attack is staged, or whether the attack is staged at control handover in the life cycle or supply chain (which is a vulnerable point and frequently used for the staging of attacks).
2. Re-categorised, combined and rewrote the original list of attacks into abstract overarching attacks in the newly defined categories, reducing the number of attack descriptions from 42 to 20. Described more specific examples of subcategories of attacks within the most abstract categories and attributed them to three different problem domains that were defined:

1. Benign (design) decisions which lead to future problems due to insufficient focus on, or insight or knowledge of possible cybersecurity problems arising from those decisions;
2. Genuine mistakes in design or implementation due to insufficient security awareness, lack of security expertise or training;
3. Malicious mistakes or alterations to change the intended functionality of components, systems, or solutions.

The attacks were then mapped to the life cycle model, giving the figure on the next page.

The supply chain risk analysis may show that there are residual risks that remain even after application of all realistic countermeasures. In that case, the stakeholders or the customers must decide what to do with the residual risk: risk reduction in business processes or in the environment, or with other measures. This research was commissioned by the NLNCSA, part of the General Intelligence and Security Service of The Netherlands. The article is endorsed by Dutch crypto industry. The full article can be found at https://foxdatadiode.com/news/

This starts with a risk analysis, for which a supply chain risk analysis model was devised

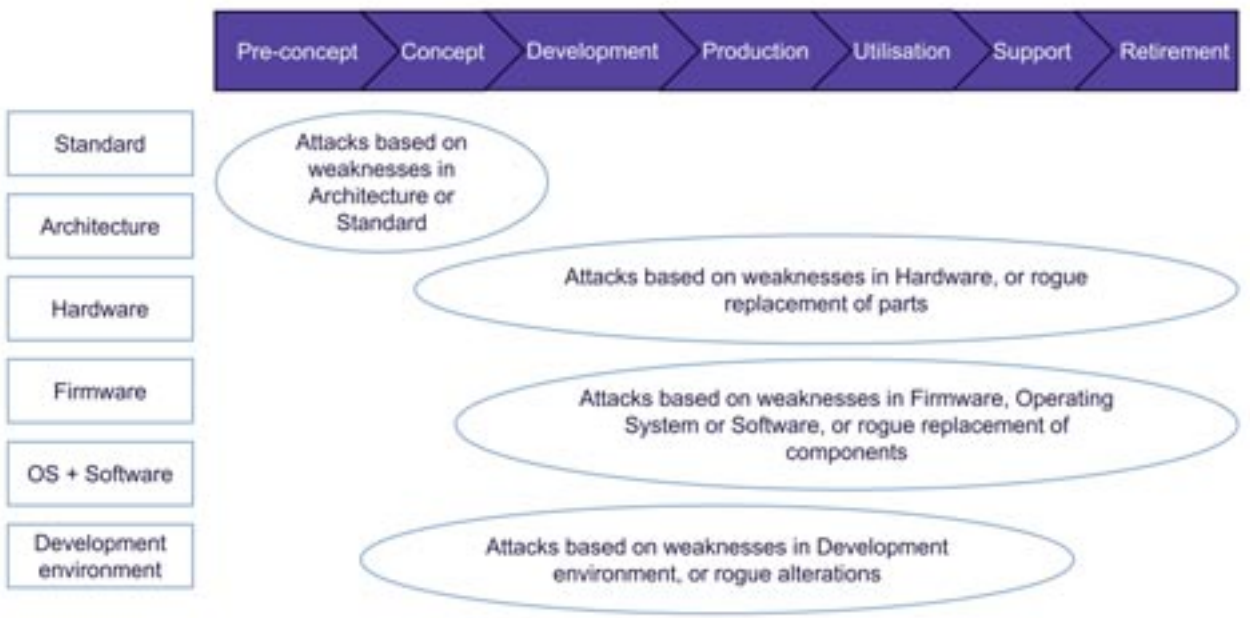


Figure 3: Life cycle with attack type categories.

Discussion and Conclusions

Supply chain attacks are a real risk, especially for high assurance products. There are multiple types of attacks that can be staged in the supply chain. To counter this problem, risk reduction is necessary. This starts with a risk analysis, for which a supply chain risk analysis model was devised. The developed model is based on a number of well-established standards and on sources of expertise, which together have led to a new model as presented in this article. In the process of updating the original model, the new model was validated in various stages and with varying groups with expertise in the field.

The model provides a basis for supply chain risk analysis that is suitable to identify possible attacks and can show how to mitigate these attacks with countermeasures. The exact implementation of the countermeasures is not part of the model.

This research was commissioned by the NLNCSA, part of the General Intelligence and Security Service of The Netherlands. The article is endorsed by Dutch crypto industry. The full article can be found at https://foxdatadiode.com/news/4