forelens

Trusting Your Software in the Age of Supply Chain Attacks

How to fix security stack blindspots
and bring detection time from months to seconds

forelens.com

forelens

Contents

Executive Summary

Current Attacker Entry Points

- *Supply Chain Attacks*
- *Exploits*
- *Human Factor*

End Game

Current Security Blindspots

- *Classic Antivirus Software*
- *EDR/XDR*
- *Firewalls*
- *Sandboxes*

Solution

Our Advantage

Learn More

forelens

Executive Summary

Securing company assets is a demanding task and mainstream cybersecurity solutions have problems to keep up with evolving threats.

Nowadays attackers are able to successfully gain access to organizations exploiting:

- trust between company and software provider, through supply chain attacks;
- software flaws, through exploits;
- human factor, through phishing and malicious insiders.

After gaining the initial access, next steps vary depending on attacker, but revolve around stealing intellectual property, company secrets, or encrypting company data to extort ransom.

Mainstream cybersecurity solutions are not designed to effectively address threats that are new, delivered by trusted source, or installed by authorized user. Detections can be suppressed by valid digital signatures, novel methods of evasion, or just mimicking behavior and communication patterns of the infected application. Some solutions will be able to provide relevant alerts, but those alerts will be lost in the flood of other false detections and insignificant events, making them hard to spot. Therefore, such attacks often go unnoticed for days or even months, causing severe reputational, financial, and operational damage.

We propose a new solution that can work seamlessly together with currently used security software and provide high quality, low volume detections based on spurious network communications performed by applications, effectively bringing the time of new threat detection from months to seconds.

What we focus on is what often goes unnoticed—should application X communicate with server Y?

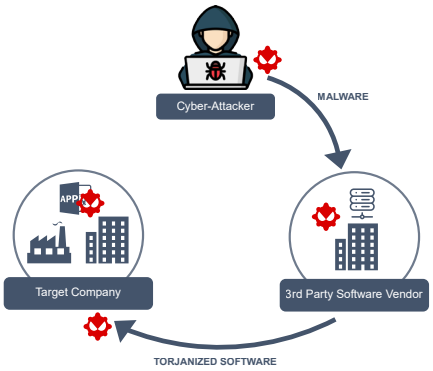
Current Attacker Entry Points

Supply Chain Attacks

"You don't attack the Pentagon, but the company that delivers them sandwiches"

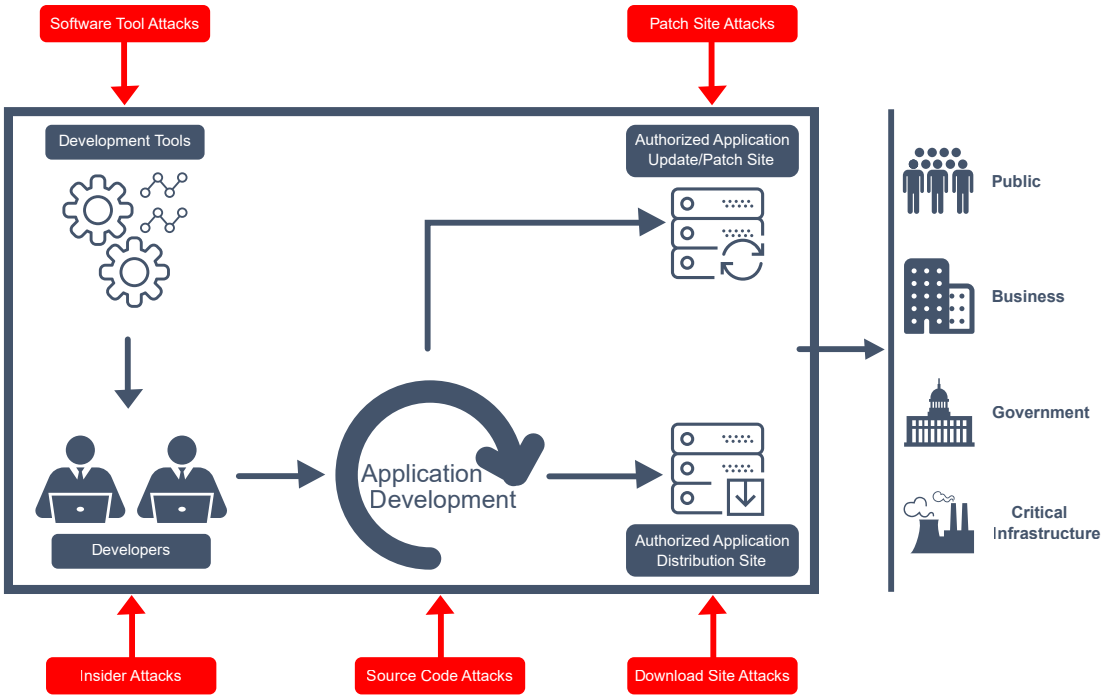
In terms of software, the supply chain refers to the whole ecosystem used to create, build and ship an application. It's not only limited to the systems inside the company providing final application, but extends to all other companies, or Open Source communities that contribute software elements included in the main application.

Successful attack on such supply chain allows injecting malicious code into the final product, which is later distributed to customers. Detection of such changes is difficult, because the final result, like application installer, software update, or library has all the marks of legitimate product—the downloaded files originate from correct domain and server, and are correctly, digitally signed. When it comes to malicious behavior, it can also be hard to tell it apart from standard functionality, as attackers try to mimic the benign activity patterns of the modified software.



Because of supply chain complication level, there are multiple places where malicious code can be injected. Methods shown below apply to all environments, so not only at application provider, but also across all environments that contribute to application's software dependencies.

Diagram below shows entry points used in attacks to date.



Current Attacker Entry Points

Exploits

"Now your application does what we want, not what you paid for"

Exploits take advantage of flaws in software. Successful attack affects behavior of application and opens a gateway that can be used to install backdoor, ransomware, or any other type of malware.

There are multiple reasons for existence of such flaws, from negligence during testing phase, low qualifications of developers leading to insecure coding, to unintentional error caused by overall complexity of application. Unfortunately, similar to supply chains, the flaw can exist in code created by software vendor, but can also be inherited from any 3rd party components used to build the application. That creates yet another problem—if vulnerability is fixed by provider of one of building blocks, when the new version will be incorporated into the final product? The answer is not easy, because sometimes upgrading to the latest, fixed version requires major changes in the main application, which due to financial, or time constraints might not be a quick operation.

Exploits can be divided into two groups:

- unknown to the public, often called zero-days, which pose the biggest threat, as neither vendor can provide fix, or advice for how to temporarily mitigate the threat, nor security software is aware of the new technique and can fail at detecting it;
- known and fixed, often called one-day, or n-day. Those flaws are publicly known, vendors already provided new, fixed versions and a multi layered security solutions should be better at detecting the attack. They are less dangerous than zero-day variants, although because of mentioned software dependencies, it can take time before update will be incorporated into the final product. Finally, there is often a delay in updating software running inside the organization, which can be caused by lack of software monitoring systems, staff shortage, or simply because of oversight.

Current Attacker Entry Points

Human Factor

"Hi, it's Jane from IT, your password is about to expire, please use this link to update it"

As mentioned earlier, attackers can exploit trust between customer and software vendor to perform supply chain attacks, but they can also exploit trust between people, or between person and company. It can be a fake email, text message, or even a phone call from IT department asking to fill a form to reset password. If it is well prepared and timed, it can let attackers in by taking over user credentials, or other secrets, allowing to bypass even a 2 factor authentication and allowing access to the company resources. Security awareness trainings do help to make such attacks harder, but even trained people can fall for a well prepared phishing.

In some cases the attack might come in form of financial gratification for an employee in return for a "favor". The insider will either provide credentials allowing access to the infrastructure, or deploy malicious software, to the same effect. Imagine a server running old, vulnerable software that is perfectly isolated from the internet, but easily accessible and exploitable from inside the company's network.

Last category originates from human attraction to convenience. Systems exposed to the internet can have weak, or no authentication, either because it's easier to reach them, or they were exposed only for testing and then forgotten. Similar things happen to development environments that often have lower protections for convenience sake, but after gaining access to them, attackers can use them to move deeper into the organization's infrastructure.

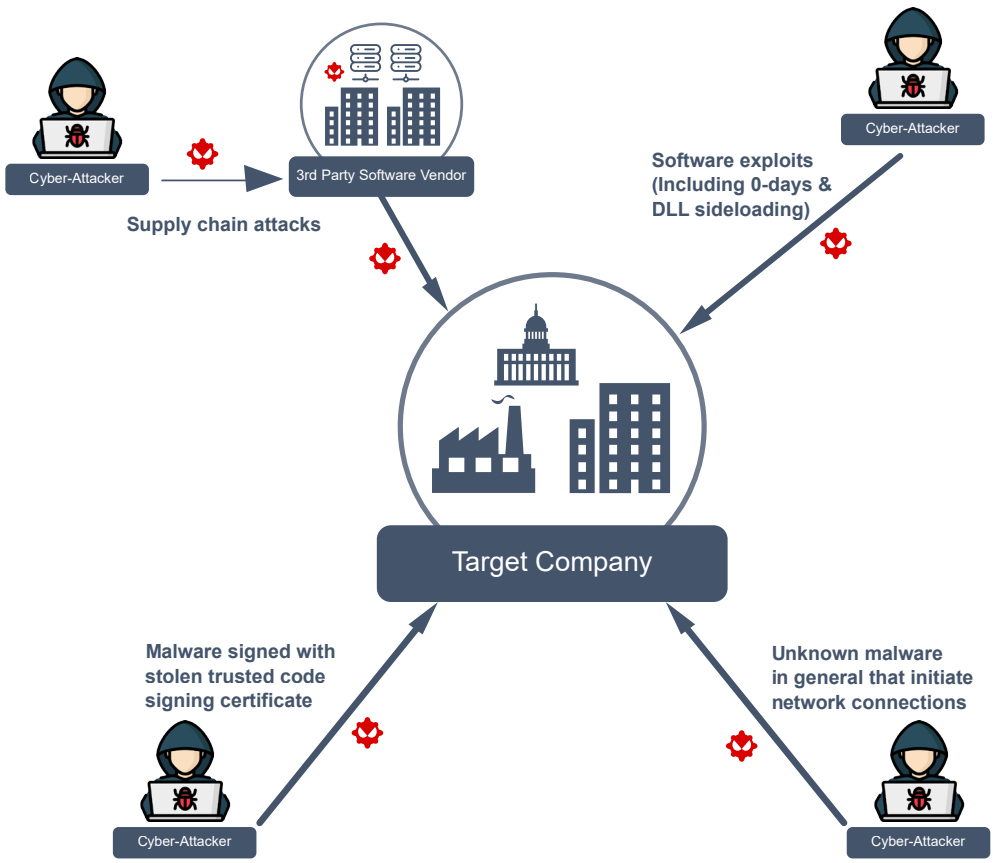
In context of software supply chain attacks, all those possible attack vectors concern not only single organization, but also all software vendors it depends on, so failing in one place will have an avalanche effect.

End Game

After gaining initial access, attackers unfold their main plan. Planting a backdoor to have persistent access to company network and deploying ransomware to exfiltrate, encrypt and hold company's data hostage are the two most common scenarios. If the target is not the company itself, attackers might just want to use the newly gained access to exploit the trust and reputation of the victim and use it as a vehicle to move into the victim's customers.

Depending on attack method, it can be either noisy and easy to detect—encrypting all the files will be noticed fast—or slow and stealthy, like quiet backdoor allowing slow data exfiltration and moving around the network to other, more interesting systems.

At this point, assuming the security solutions are in place, there is a chance that deployed malware, or its communication will be detected and blocked, but for current mainstream products it's not an easy task, which will be discussed on next pages.

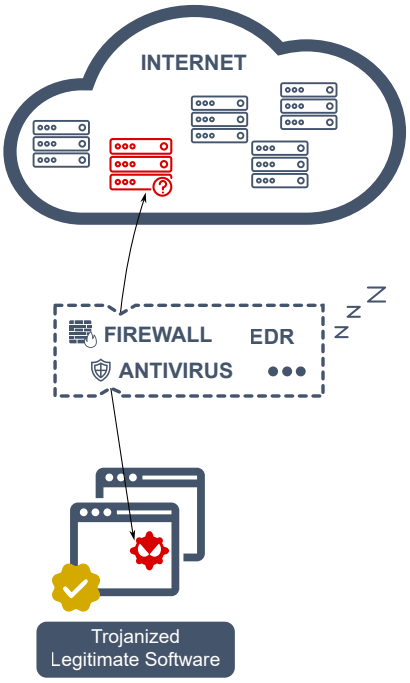


Current Security Blindspots

"If it quacks like a duck, walks like a duck..."

Classic Antivirus Software

Are great at detecting known and similar to known threats, problem begins when attack is new, like a 0-day exploit, or comes from trusted, digitally signed application. For performance reasons applications assumed to be trusted can be exempted from file and behavior analysis, which is dangerous when it comes to supply chain attacks.



EDR/XDR

Excel at collecting data about everything that happens in the system, the problem arises when security team has to correlate, interpret and extract the important parts from all the noise. The built-in information filtering looks for anomalies and is based on artificial intelligence verdicts, or a set of hand made rules, but still the final result requires a team to monitor it.

Firewalls

Filter network traffic based on blocklists plus anomaly detection, which means they can have similar problems when application is trusted and pretends to operate in a standard way. The deep inspection of every data packet flowing through network creates higher hardware requirements, that's why firewalls working locally on computers must make some trade-offs, for example digitally signed applications are by default deemed trusted. Specialized firewall appliances allow complete inspection of every packet, but lack the context - which application, running in what type of environment did initiate the connection.

Sandboxes

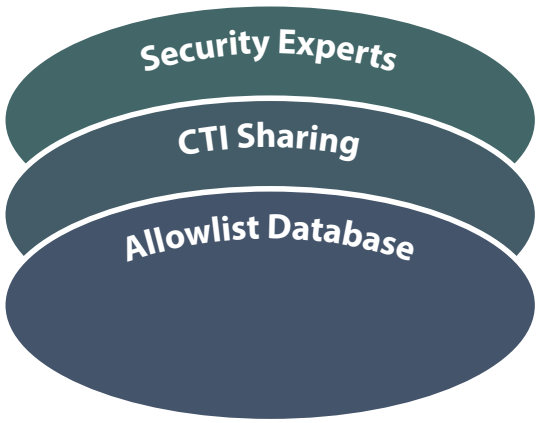
Focus on detecting threats that can be missed by a firewall, or antivirus alone. They pretend to be a normal computer and quietly observe if the analyzed application doesn't behave in suspicious ways. Their main problem is that they can't run analysis forever, so if nothing happens during the first 1, or 2 minutes, they'll move on to analyze next file. Moreover, the environment used to analyze files is only an imitation, which means that malware might try detect it and refuse to perform malicious actions. For the same reason, malware that is targeted at specific group of computers will not execute, because the fake environment won't have all the required characteristics.

Solution

"Don't guess, use knowledge"

The common denominator for all the attacks is a network connection going out, back to attackers. It can be initiated right after start, or after some time of fake inactivity, but the planted malware needs network communication to receive commands, download more malicious tools, send the encryption keys it's about to use to encrypt documents, or exfiltrate data. Applications used across organizations have well defined, but hardly ever published, lists of domains they need to communicate with. Current raise of supply chain attacks made SBOM (Software Bill of Materials) a popular tool to provide easy overview of building blocks inside application, in similar fashion we propose SBOC (Software Bill of Connections), which transparently describes which connections are required by application components.

Currently there's no unified process of documenting and enforcing networking requirements for software. Firewalls with allowlisting, while being the strongest type of network protection, are not very popular. Building and maintaining such allowlists requires dedicated team and financial investments and every organization maintaining such lists does it independently, which means that a single application that is used in 10 companies will consume 10 times more work.

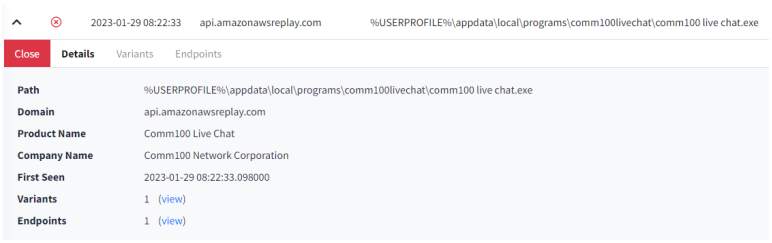


Forelens provides an SCDR (Spurious Communication Detection and Response) system, which is a new cybersecurity approach to modern threats. It consists of unique, first to market allowlist database that is curated by our security experts, allows easy Cyber Threat Intelligence sharing across all our customers and provides a quick answer to the question—"should application X communicate with server Y?". Our lightweight agent gathers information about connections and their context, allowing us to detect unexpected communications performed due to supply chain attacks, exploits, or unknown malware. The solution can seamlessly work together with other security solutions already in use and provide detections that would be otherwise missed.

Our Advantage

2022 - Comm100

Attackers modified the application installer which was later distributed from legitimate vendor's servers. The malicious version was properly signed with vendor's certificate, thus was seen as trusted by security software. The malicious addition was communicating with attacker controlled server, which qualifies as spurious communication and is reported as such by our system.



2021 - Log4j exploit

Vulnerability in Log4j library that is a common building block of many applications. The flaw was discovered in 2021, but was introduced into the library in 2013. Because of the library's prevalence, it is expected that attacks using this vulnerability will appear over next years. The security companies responded very quickly by releasing updates to block exploitation attempts, but attackers already try to come up with methods to bypass the protections. If successfully exploited, it allows execution of any code on the targeted computer, but whenever the code will need to communicate back with the attacker, it will be recognized by our system.

2020 - SolarWinds

Attacker gained access to SolarWinds infrastructure and modified it's code. As a result, the malicious addition was distributed as a legitimate update to all customers.

The attack, affecting at least 100 private sector agencies and 9 federal agencies, took 9 months to be discovered.

It was correctly, digitally signed, distributed from official vendor's servers, while communicating with attackers it was mimicking network protocols used by the infected application. In case of sandboxing, the malicious payload would not trigger, as it was designed to stay dormant for many days after installation. The only giveaway was the target of the network connections—a domain used by attackers, but never used by the software in normal conditions.

2019 - Asus

Attackers managed to add a backdoor to the LiveUpdate, which is preinstalled on Asus computers. Modified version was distributed as an update to Asus machines all over the world. An estimated 500 million devices downloaded and executed the malware. As in SolarWinds case the digital signature and the source of update were benign, so no alerts were raised by security software. Moreover the backdoor was targeting particular set of computers, based on their hardware ID, so it would not execute in sandboxes. On the targeted computers, the only difference between standard and malicious version would be an additional, spurious connection reaching out to attackers.

Learn More

To learn more about modern threats and how we can help to detect them, visit our page and blog at https://forelens.com.

To request a product demo, contact us at demo@forelens.com.

Together we can bring down detection of threats from months to seconds.


forelens
forelens.com