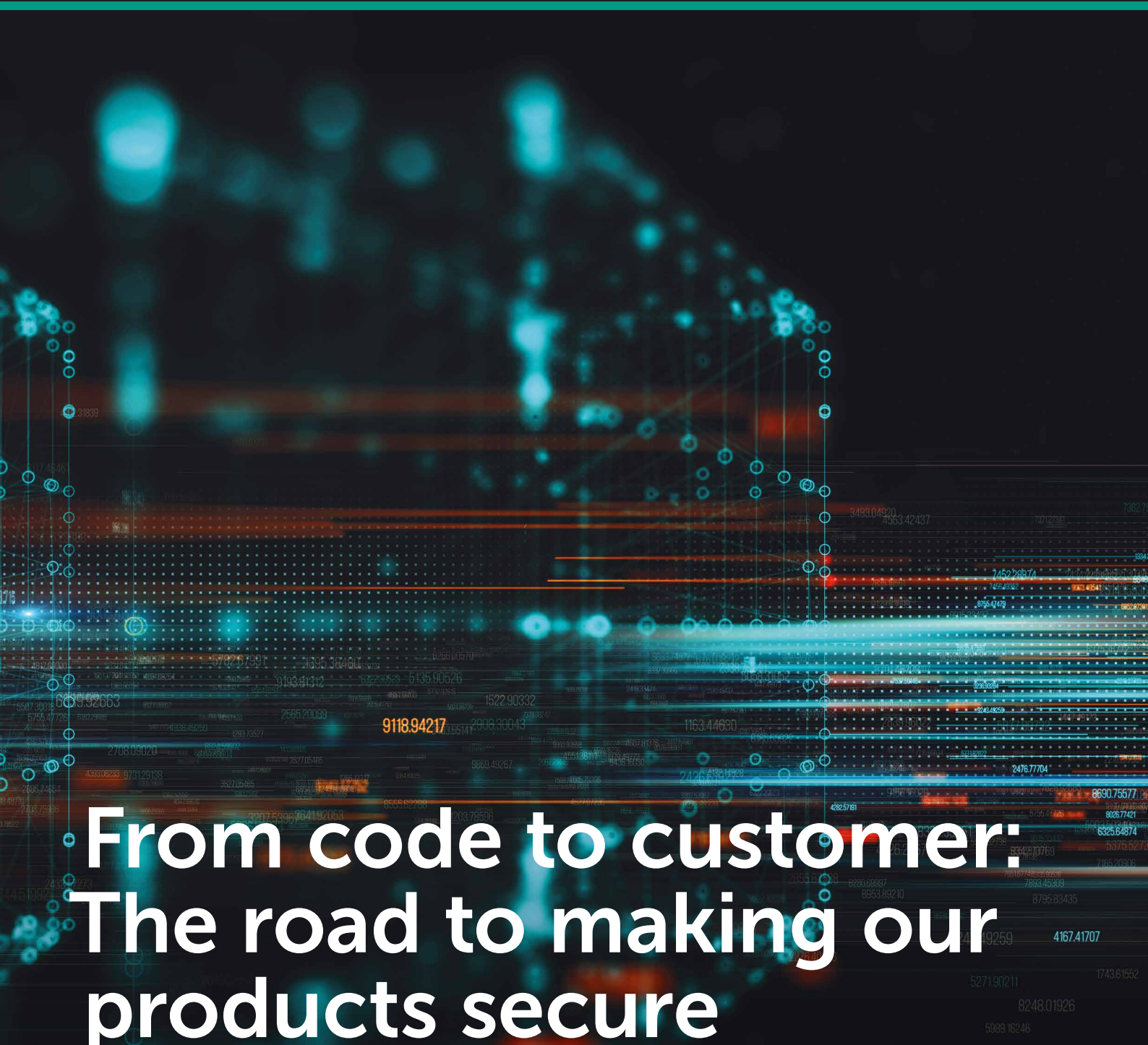


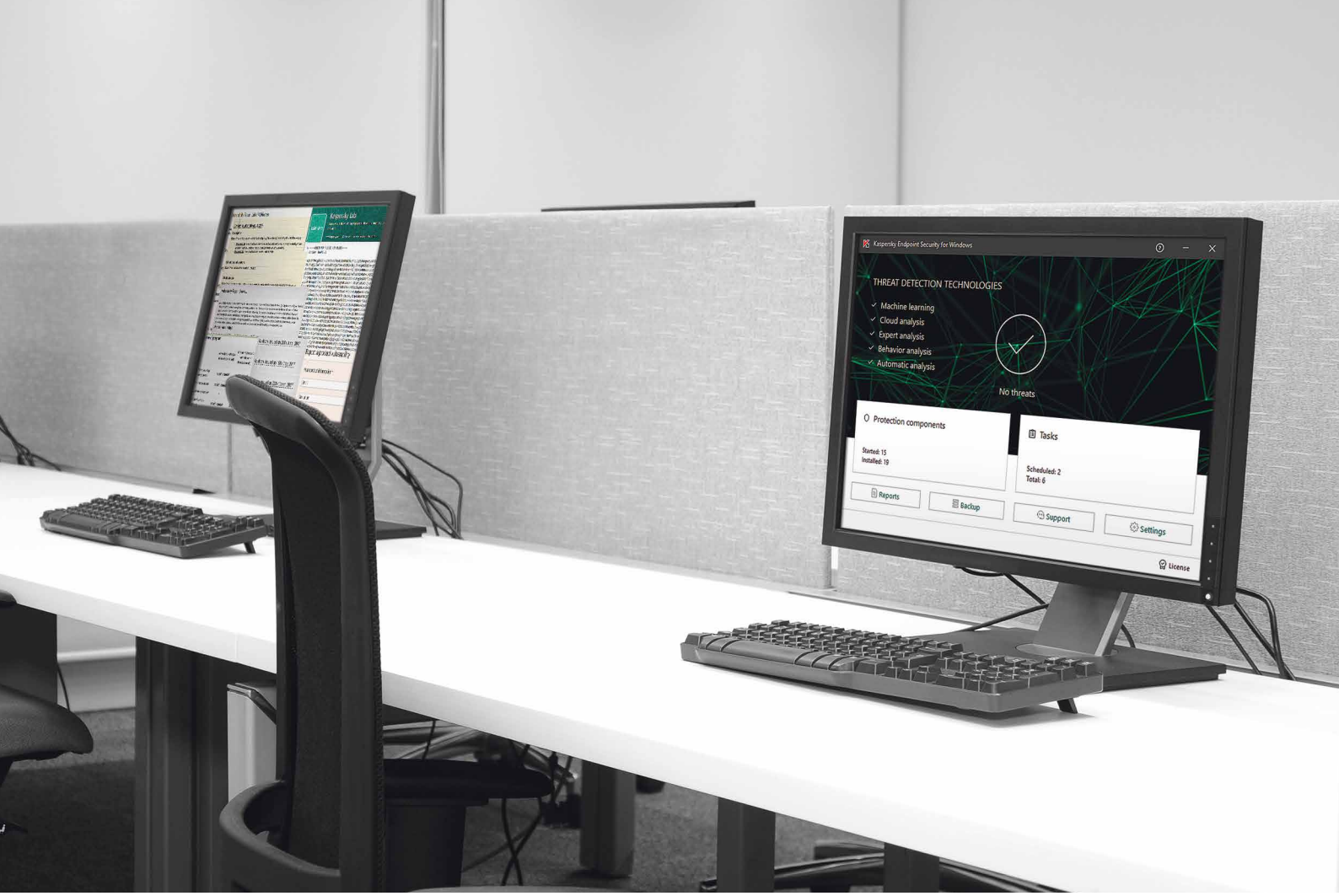
From code to customer: The road to making our products secure

Executive summary

As the world becomes more and more interconnected, global reliance on safe, trustworthy technology intensifies and the security of products themselves becomes increasingly important. The evolution of security products takes place in parallel with the growth of the vulnerabilities research industry, and as the threat landscape develops, the structure and nature of our products become more complex too.

Software development is a multifaceted process, with many stages in the road from code to customer. Unlike most other software, security products consist of numerous components, some deeply integrated into the OS and this makes solving security issues even more important – not doing so could take an entire system down.

One of the biggest challenges in software development is avoiding what developers refer to as 'whack-a-mole', an ongoing situation where the same problems keep reoccurring, again and again. This is inefficient – and dangerous.



When the architecture of software components is done right, this situation can be avoided. And combined with a collaborative software development lifecycle that puts security front and center of every single step in the development process, it's even safer. This is what Kaspersky does – we've adopted a fundamental, strategic approach involving cross-team collaboration, diverse internal and external information sources and ongoing education.

We've refined the entire process and the result is hands-down the best way to embed safety into our products – and deliver the same to our customers. Continue reading to find out how we do this.

From code to customer: The road to making our products secure

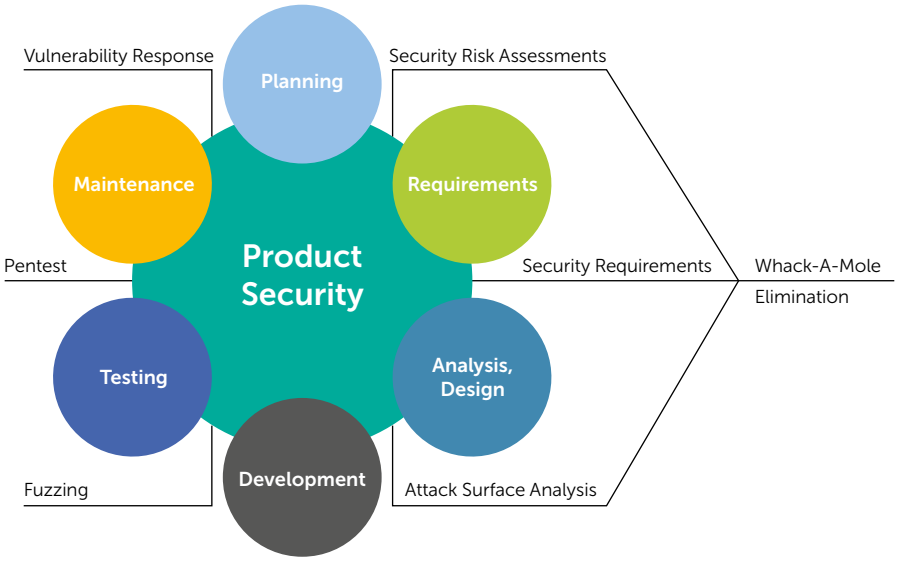
We at Kaspersky Lab take our role as a leading security vendor seriously. In the process of developing our products, we are guided every step of the way by the core principles that make them secure. Unlike other software, security products consist of many components, some integrated deeply into the OS, which is why solving security issues in our products is so important.

It's been over two decades since we developed our first antivirus solution, and over this time we've gained unique, first-hand experience in how to respond quickly and effectively to the new and evolving challenges of the cybersecurity industry. The evolution of security products – as well as other software – occurs in parallel with the growth of the vulnerabilities research industry. As the threat landscape continues to develop, the structure and nature of our products become more complex, with enhancements and new features in every release.

We understand the direct link between making a product more complex and the number of potential vulnerabilities to be found in it. Our specialist product teams and our entire development process are geared towards ensuring that our software engineering processes are as safe as possible. Building the highest levels of security into our products is at the heart of what we do.

It's all in the foundation

When the architecture of software components is done right, a 'whack-a-mole' scenario – where problems keep recurring – can be avoided, and the same applies to fixing vulnerabilities. Our product teams work closely with our product security team to ensure that our architecture is secure.



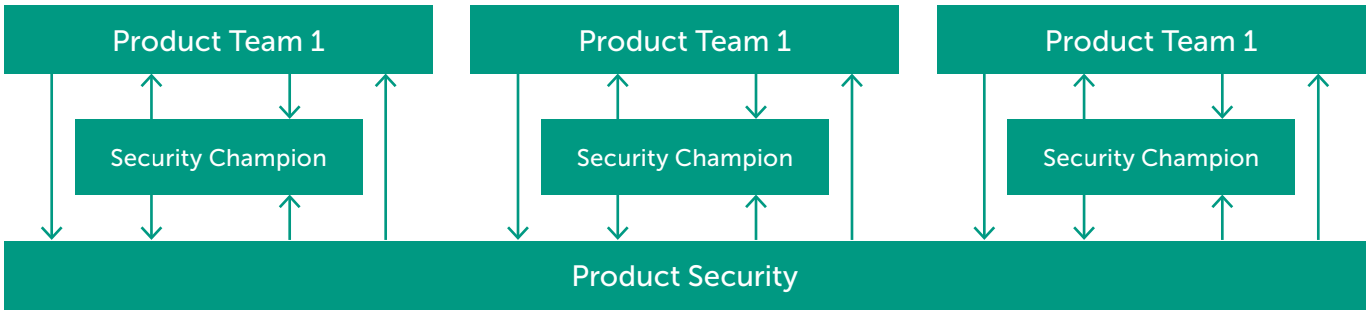
Securing the development process. The first three steps in particular – planning, requirements and analysis and design – help to avoid a 'whack a mole' situation.

This approach has allowed us to avoid the problems highlighted by Google 'Security Princess' Parisa Tabriz in her keynote at Black Hat USA 2018. In her address, Tabriz discussed an issue that all developers are familiar with – the frustration of receiving reports about security vulnerabilities that have been previously fixed, or are a trivial variant of a known bug or even a symptom of an underlying condition or process failing that was known about but wasn't addressed.

At Kaspersky Lab, this is where the Secure Development Lifecycle comes in, an approach which involves making security a priority during the development process of a product. Eliminating the likelihood of a 'whack-a-mole' scenario during architecture development not only eradicates ongoing problems with the same vulnerabilities, it also frees up resources that can be redeployed to work on developing other products and maintaining already released products. The net result is secure product architecture that has numerous advantages, including:

- Minimizes vulnerabilities that can lead to serious architecture changes at core components during the maintenance step of development
- Frees-up resources during maintenance
- Reduces the amount of security updates necessary.

How the product and product security teams interact. The security champion is an additional layer that ensures that information from product security is correctly applied.



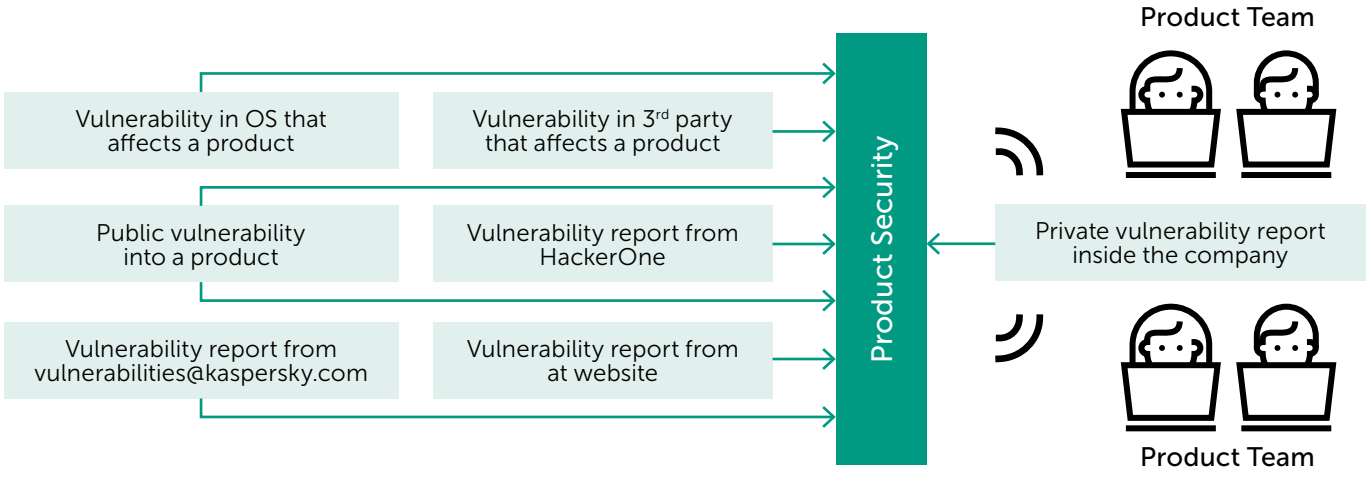
Our product security team is the entry point for R&D for all issues relating to the security risks of products and infrastructure. The team is responsible for a number of critical tasks, including preparing the initial requirements, code auditing, vulnerabilities response, risks analysis, vulnerabilities assessment, providing mitigations, fuzzing processes integration, penetration testing, and more. Our Anti-Malware Research Team (AMR) also provides crucial input during these steps.

Sources of vulnerability information

Even after the product team has invested significant time and effort into writing software according to security standards, an attacker can find a weak spot inside the product or its environment (OS), which can be compromised. For this reason, we use diverse sources of information about vulnerabilities to optimize our product security.

We recognize various sources of information for security risks and vulnerabilities: Publicly disclosed vulnerabilities in an OS that may be affecting our products in some way, vulnerabilities in third-party libraries or software which we use in our products, public vulnerabilities reports from researchers, reports from our Bug Bounty program portal, reports which hackers send to our vulnerabilities mailbox (vulnerabilities@kaspersky.com), reports which researchers submit to our online form and privately reported vulnerabilities from our security researchers, architects and penetration testers.

The different sources of vulnerabilities intelligence that Kaspersky Lab receives

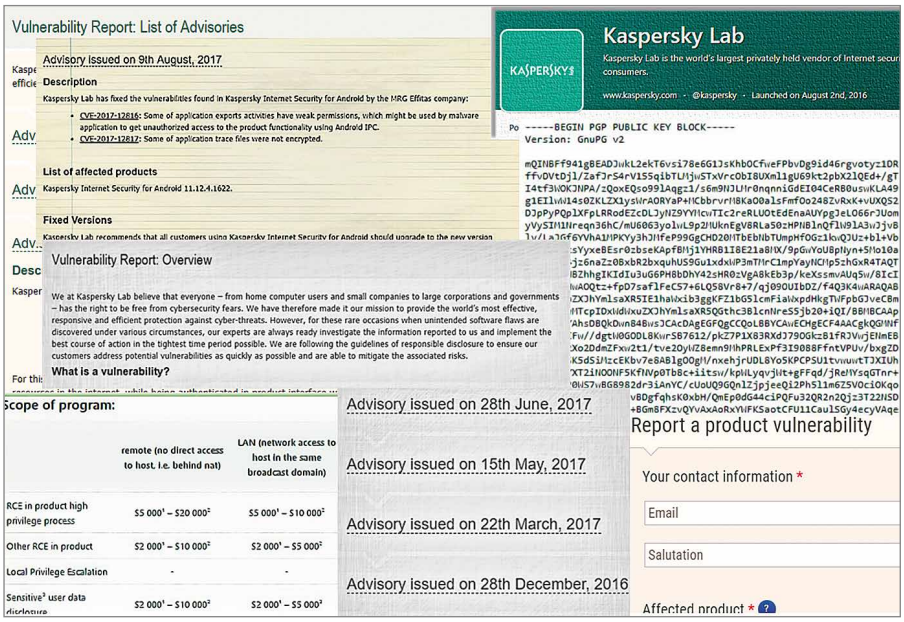


Publicly disclosed unpatched vulnerabilities in our products are critical because by their nature, attackers can use them to attack our customers. Depending on the CVSS (Common Vulnerability Scoring System) score and the potential impact level on customers, fixing these vulnerabilities is our top priority. Fixing publicly disclosed vulnerabilities into operating systems and third-party software that affect our products is also a high-priority scenario.

The HackerOne vulnerabilities reporting platform gives us a flexible and regulated way of accepting reports from researchers. The platform's workflow for vulnerabilities reports processing begins from triage and includes compensating 'bug bounty' to researchers and addressing around the disclosure of sensitive information about a particular vulnerability. (A hacker can request complete or partial information disclosure about a vulnerability after its fixed and the fix has been released to the public.)

Some vulnerability reporters prefer to submit their discoveries directly to us via our **dedicated vulnerability reporting mailbox** at vulnerabilities@kaspersky.com. We recommend using encryption to submit this sensitive information using our public PGP key, a common practice in the industry to secure this type of process.

Sources of vulnerability reporting



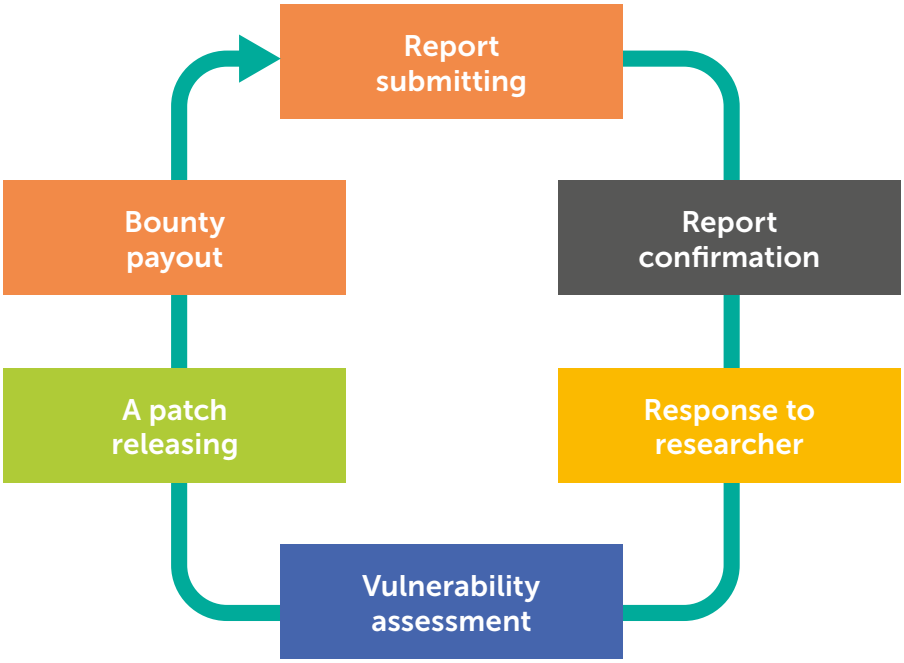
Responding to reports

Responding to vulnerabilities reports is a separate process with a special workflow. The response process starts when we receive information about a potential vulnerability that may affect our products. Sometimes, we receive a report that's supposedly about a vulnerability in one of our products but is in fact a vulnerability or weak spot inside an OS. In such cases, we also provided an update to mitigate this situation.

Kaspersky Lab's involvement with the Bug Bounty program at HackerOne is in response to the evolving challenges of the security industry. Of course, we have our own highly specialized security researchers and architects in-house, but we also recognize that external, independent security researchers can bring their own points of view and thinking to the process, and by letting these external researchers pentest our products, it makes them even stronger and more secure.

We also believe that paying hackers to report vulnerabilities is right as well as smart – not doing so may see them being tempted to sell their information to cybercriminals instead. Their qualified reports must fulfil stringent guidelines, including a detailed explanation of the vulnerability they're reporting together with technical details and an example of a reliable working exploit or proof of concept.

Vulnerability processing workflow in a bug bounty scenario



The scope of our Bug Bounty program currently includes Kaspersky Internet Security 2019 Beta and Kaspersky Endpoint Security 11. Vulnerability exploitations should apply to Windows 8.1+, and we compensate researchers for discovering Remote Code Execution (RCE) vulnerabilities, Local Privilege Escalation (LPE) and Information Disclosure (ID) – the latter being limited to sensitive user data like passwords, payment data and authentication tokens. For a detailed report and working exploit example of the <<unicorn>> vulnerability, which allows an attacker to remotely execute malicious code inside our high-privilege process using the 'man-in-the-middle' vector, the pay-out can be as high as $100k.

	Ring 3 (low privileges)	Ring 3 (high privileges)	Ring 0	Ring -1
AV bypass				
Information Disclosure (ID)				
Local Privilege Escalation (LPE) (Escalation of Privilege)				
Remote Code Execution (RCE) vulnerabilities				

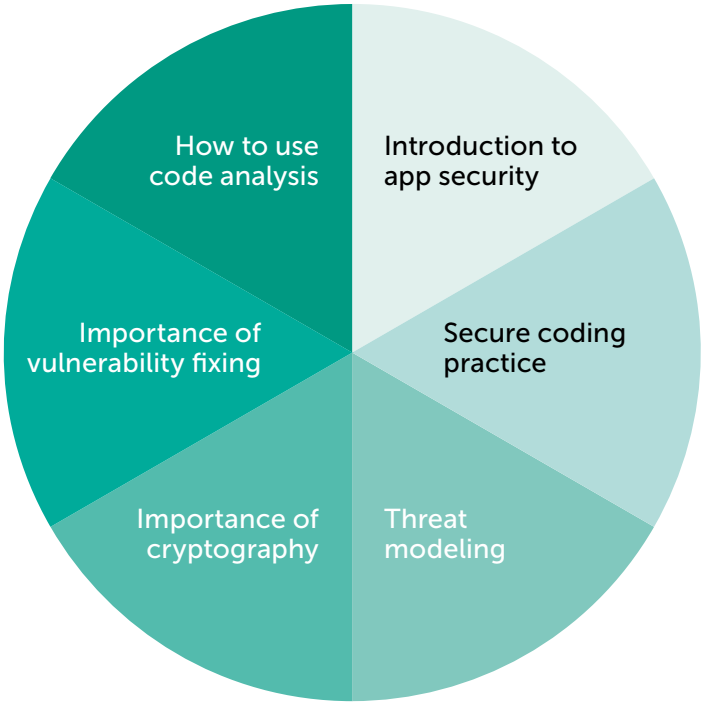
Types of vulnerabilities found in antivirus products. Attackers use 'AV bypass' to circumvent major parts of a product's protection. The darker the cube, the more serious the level of danger

- User mode (**Ring 3**) is where the antivirus service and GUI processes execute – the antivirus service is a high priority process, while GUI is low priority.
- Antivirus drivers execute into **Kernel mode** (**Ring 0**) and obtain access to the Windows kernel and all processes in VM (virtual memory). This kind of exploitation is highly dangerous because Ring 0 has the most privileges and a successful exploitation here can lead to the entire system being compromised.
- Some antivirus products contain special components that work at the hypervisor level (known as **Ring -1**). These components control cross-VM (virtual memory) operations and provide protection from screenshotting. Exploiting a vulnerability at this level compromises virtual machines and can also bypass built-in security measures (such as Device Guard and Credential Guard).

Never stop learning

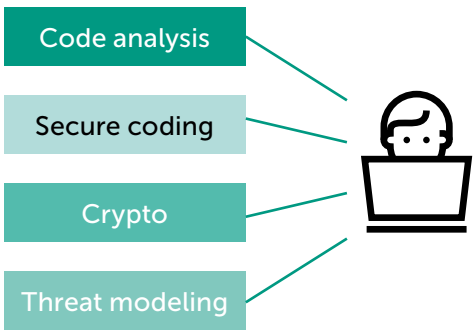
As we continue to integrate secure practices into our software development lifecycle, we also encourage our developers and architects to learn about them. We combine our knowledge of secure development from our existing R&D experience with insights from our product security team who continuously update their knowledge and skills with new and emerging practices. The result is the implementation of the following major vectors into our product security training:

Security training for developers, engineers and architects



This approach is highly beneficial to our developers and to the company as a whole - please refer to the image below for more information. It not only encourages developers to think about different areas of computer security that they may not otherwise consider and it also helps reduce the total cost of software maintenance and boosts our reputation.

Advantages of ongoing developer training

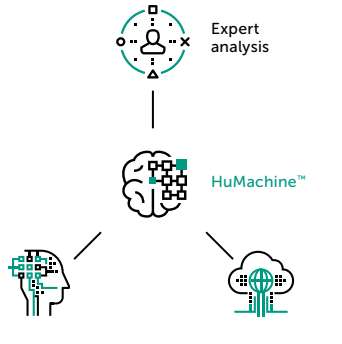


Advantages

- Lower expenses for security updates issue
- Freeng human resources for products support
- Improving the reputation of the company
- Improving developer's skills

Putting security front and center of everything we do

The process of creating secure software isn't easy, especially when it comes to complex products and solutions. To deliver optimum security levels, security should be front and center of every stage in the software development and maintenance process, including writing security requirements, risks assessments, attack surface analysis, fuzzing, pentesting, vulnerability response and developer education. And at Kaspersky Lab, this is exactly what we do. The result? The world's most tested, most awarded security.



Kaspersky Lab
Find a partner near you: **www.kaspersky.com/buyoffline**
Kaspersky for Business: **www.kaspersky.com/business**
Enterprise Cybersecurity: **www.kaspersky.com/enterprise**
IT Security News: **business.kaspersky.com/**
Our unique approach: **www.kaspersky.com/true-cybersecurity**

#truecybersecurity
#HuMachine

www.kaspersky.com