ACROS Application Security Analysis

Frequently Asked Questions

1. What is ACROS Application Security Analysis?

In a nutshell, ACROS Application Security Analysis is an analytic project aimed at discovering as many vulnerabilities in a software product as possible using an agreed-upon amount of effort. Using this service, software vendors and online service providers can significantly cut down their costs of developing, testing and distributing security patches, avoid damage to their reputation, keep their customers' data safe and their critical services secure.

2. What is the purpose of ACROS Application Security Analysis?

It is usually in the best interest of software vendors and online service providers that their customers and independent security analysts don't find security flaws in their products, as this damages their reputation, hinders negotiations with prospective customers or, in the worst case, allows malicious individuals to attack their systems or their customers' systems and damage their business and sensitive information.

ACROS Application Security Analysis helps minimizing the likelihood of that happening. By having a team of skilled, knowledgeable and motivated security analysts hunting for vulnerabilities in their product, a vendor is able to fix the dangerous vulnerabilities before anyone else even knows about them. Furthermore, by preemptively fixing them, the chances of these vulnerabilities ever becoming known are significantly lowered or altogether eliminated.

Finally, our security analysis helps our customers in obtaining security certifications such as PCI/DSS or ISO 27001.

3. What aspects of a product does ACROS Application Security Analysis cover?

Our security analysis can cover all aspects of the product itself and its deployment in a production environment. We can analyze the product's **architecture**, focusing on security errors that will surely be much more expensive to fix once the product has been implemented. When implemented, we can analyze the product's **implementation** to find vulnerabilities in its **source code** or **binaries** in a "white box", "black box" or "grey box" manner. We look for all known types of security errors from different points of view, for example: how a malicious attacker could exploit them for gaining access, how a malicious administrator could decrypt users' encrypted data or even how a legitimate user could try to bypass a security policy in order to make his life easier.

We analyze the product's default **installation** and **configuration**. We review the product's **installation guides**, administration and user **manuals** to uncover ambiguities, potentially dangerous instructions or information that might result in insecure installation, administration or use of the product.

4. What are your deliverables in an application security analysis?

You will receive a 3-5 page **security update** every time we discover a vulnerability, as soon as we verify and document it. Along with the security updates, we'll send you any **custom-made tools** (including the source code) needed for reproducing the problems or understanding the analysis.

At the end of the analysis, we'll provide you with a detailed **final report**, which incorporates all prior security updates, and contains other project data, various statistics of the discovered issues from different points of view, and high level conclusions and recommendations.

We present the results of the analysis in two conceptually different presentations. The **executive presentation** is targeted for managers and presents the findings from their point of view. It focuses on the impact of the discovered vulnerabilities on their company based on the worst-case scenarios. The **engineering presentation** presents the vulnerabilities from the developers' point of view, focusing on the security errors that caused them. Its aim is to raise awareness of those types of security errors that have resulted in the most severe security issues – so that such issues would not appear again in future products.

5. What types of security issues do you look for?

In ACROS Application Security Analysis, we look for both **generic** and **specific** security errors and attacks. Generic security errors are the ones that are well known and documented in security books and research papers, for example: buffer overflows, SQL injections, cross site scripting, format string errors and weak random number generation. Specific security errors are unique for a particular product as a result of its unique functionalities and requirements. While every good security analyst should be well familiar with generic security errors, vulnerabilities and attacks, it takes a rare talent and ways of thinking to be good at finding product-specific security issues. We're proud to say that our experts are very efficient at finding both generic and specific security issues.

6. What types of applications do you analyze?

We analyze all types of applications: from mobile application, web applications, Windows/Linux desktop applications, servers, clients, databases, hardware drivers to embedded applications and JavaCard applets, language interpreters and virtualization solutions, regardless of the operating system they're running on.

7. How many vulnerabilities do you find on average in a typical analysis?

In a typical analysis of a "security mature" software product we discover on average one critical security issue every 2-3 researcher days. By "security mature" we mean that the product was designed and implemented by engineers trained in secure coding and familiar with most of the security errors and vulnerability types. Also, a sufficient amount of effort needs to be invested in the analysis to reach this ratio, which is determined by the typical "vulnerability arrival curve".

8. What is the typical vulnerability arrival curve?

Similar to functionality testing, there is a recognizable pattern of vulnerability arrival in our security analyses. Figure below shows the different phases of the security analysis process with regard to the vulnerability arrival rate (blue line). The orange line shows the analysts' level of knowledge about the product, growing throughout the analysis.

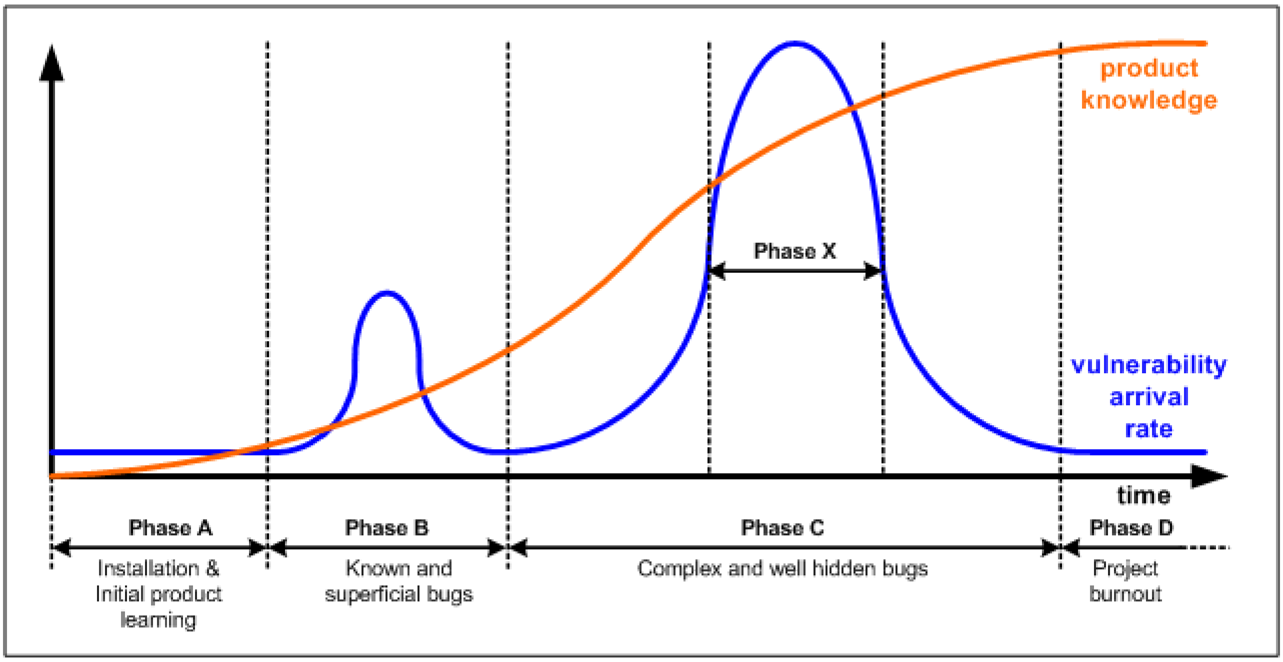
Phase A is the initial phase, where the product is first installed, its documentation read and the analysts start to get familiar with the product's functionality. In this phase, very few vulnerabilities (if any) are found. The analysts' knowledge about the product is low during this phase.

In phase B, known vulnerabilities (publicly known, known from past projects or similar to either) and "superficial" vulnerabilities (those easier to find even without deep insight into the product's implementation) are found.

Phase C is where the analysts' knowledge, gathered during the previous phases, starts to result in discovered vulnerabilities that would be difficult to find without deep product knowledge. This is the phase that usually produces the most hidden, and often the most severe security problems.

Phase X is a sub-phase of phase C, in which the vulnerability arrival rate is extremely high. This is the phase that usually produces the most value for the money.

In phase D, the analysts' product knowledge is very high, but they cease finding new vulnerabilities, as these were exhausted in phase C. It makes little sense to continue the analysis once phase D has been reached.



9. Do you first find more easily discoverable issues and later more hidden ones?

Not necessarily. While it might sound intuitive that a security analysis would at first produce more easily discoverable vulnerabilities ("low hanging fruits") and only later on more hidden and complex ones, the reality is not that simple.

While the vulnerabilities found during Phase B (see the vulnerability arrival curve above) are usually highly discoverable – i.e., very likely to be found by others -, Phase C often reveals additional highly discoverable issues.

Most vulnerabilities can be found in more than one way. For example, a cross-site scripting issue can be found by sending malformed data to the web server, by reviewing the source code or by analyzing the network traffic – or a combination of all three. Some issues of this type will be discovered relatively quickly by putting special characters to the URL arguments, while others will only be found from the source code.

10. How do you evaluate discovered security problems?

Once we find suspicious behavior or code in a product (we call that "a lead"), we analyze it to determine whether it could lead to a vulnerability. Sometimes, a lead turns out to be a non-issue, other times it evolves into a potential or an actual security problem. A potential security problem is an issue that doesn't currently present a vulnerability, but could easily become one under certain circumstances. For instance, using a cryptographically weak random number generator for message encryption would be a security issue in a cleartext communication, but using SSL for encryption makes this issue non-exploitable. Nevertheless, we would document and report this issue as a potential security problem, for it might easily become exploitable in a future version of a product – or in a product reusing this product's code.

For every actual security problem, we write a step-by-step **demonstration** for reproducing it, an in-depth **analysis** that helps the developers understand its background and impact, worst-case attack **scenarios** and

recommendations for eliminating the vulnerability. A **summary** helps managers quickly assess the severity and impact of the issue and properly prioritize and delegate it. **Mitigating factors** describe negative factors on impact or exploitability of the security problem. The vulnerability is placed in one or more **error categories**, helping managers detect the types of errors most frequently present in the product – and take steps for eliminating them in the future. We identify the **components affected** by the vulnerability, making it easier for managers to assess each component separately from the security point of view.

Furthermore, for each attack scenario we assess its **impact**, **required access** (access required for performing the attack), **configuration dependence**, attack **simplicity**, **cost** of the attack, **detectability** (likelihood that the attack would be detected) and **traceability** (likelihood that the attacker would be identified). The overall **severity** is calculated from these assessments, ranging from very low to very high.

We also make an assessment of the **discoverability**, i.e., the likelihood that someone (besides us) would eventually find the particular vulnerability.

A compound value, **exposure**, is calculated from severity and discoverability.

A typical security problem report is a 4 to 5 page document delivered in Adobe PDF, Microsoft Word or HTML format.

11. Do you use automated tools?

We use numerous tools in our analyses, most of them developed by our analysts as a result of the specific requirements of our research. Tools can easily find some types of vulnerabilities, but most non-trivial vulnerabilities can only be found by intelligent human minds.

We combine automated and manual approach to get optimal results. It should be noted, though, that automated security analysis is only efficient at finding very simple and highly discoverable vulnerabilities, making it a very good "hygienic" tool. Our experience shows that "security mature" products (see question 7) contain very few, if any, vulnerabilities that can be caught by commercial security scanners. Our analysis, on the other hand, still uncovers numerous highly severe vulnerabilities in these very products. One of our recent security analyses yielded 31 vulnerabilities, while a leading automated security scanner was only able to find one of them, and even that with significant help and a custom-made rule.

12. How long does an application security analysis take?

It depends on the product's complexity, code size and functionalities. Products with many security-related functions (e.g., a firewall, identity management software, authentication product, e-banking solution) require more analytic effort than products playing a less significant security role. We assess the amount of optimal effort needed for the analysis based on our past experience, which in average suffices for getting to Phase D on the vulnerability arrival curve. As an example, a typical client-server multi-user product is **thoroughly** analyzed in about 80-100 researcher days. (If budget does not allow for that, we set focus areas with the customer to limit the scope and get the maximum possible value.)

13. Why do some other providers need significantly less time for a security analysis?

First, they may be using automated tools only, which severely limits the results to few easily discoverable vulnerabilities. While it's a good idea to use tools – and we use them as well – they are no real substitute for a review by skilled, knowledgeable and motivated human minds. Furthermore, tools can generate only a fraction of highly useful information we provide in our security updates and final reports, such as the in-depth technical analysis, worst-case scenarios, and severity and discoverability assessments. If you are promised to get results in a week, you can be fairly sure that your product will mostly be analyzed by automated tools. Find out more about such tools in question 11.

Second, some companies put a large team of analysts on a project to provide shorter project timeframe. Unfortunately, it takes a deep understanding of a product to find some of the more complex and hidden vulnerabilities, and this requires an analyst to learn the inner working of the product in sufficient detail. As a result, five analysts working for two weeks will not end up understanding the product nearly as well as two analysts working for five weeks, and will surely find fewer vulnerabilities.

And third, not all security experts are equal. We have been building and updating a knowledge base of vulnerabilities and attack methods for various platforms, frameworks and languages for over a decade, which provides a *very* long (and still growing) list of security errors to check for in any type of application. While most experts in the field have presentable knowledge of the common vulnerability types and attacks like cross-site scripting or SQL injection, our knowledge base goes far beyond that – all the way to widely unknown (but not infrequent) security errors that most people in the industry are not familiar with. It is not surprising, then, that utilizing our extensive knowledge base should require more effort.

14. Which phase of a product lifecycle is optimal for a security analysis?

It is best to analyze the product's architecture during, or at the end of, the design phase when little or no code has been written. It is very likely that the elimination of a serious architectural security issue will require significant changes to its implementation.

Optimally, the implementation should be analyzed before the product reaches the market, allowing you to fix the discovered vulnerabilities before they're distributed to your customers. The Beta testing phase is often ideal for this type of analysis, as the product already has most of its functionalities implemented and the bug-handling effort is already allocated due to the expected beta customers' feedback. It is not advisable to subject a pre-beta product version to an implementation security analysis, as much of the code is not implemented yet or is not in its final form. It's important to be aware that every post-analysis code change can introduce new vulnerabilities to the product.

In case the product is already on the market, it is still beneficial to subject it to a security analysis. Every security issue we discover and report to you will give you a chance to fix it before someone else finds it. Alternatively, if someone is already silently exploiting a security issue in your product against your customers' systems, our discovery of the same issue will allow you to cut those attackers off and protect your customers.

15. Do you also analyze product patches, updates and new versions?

A security analysis or a product is only up-to-date until the product changes. Today, software products change quite frequently as patches or updates are applied, or new versions are released. Each product change inevitably invalidates a part of the prior security analysis, and potentially introduces new vulnerabilities.

We provide **differential security analyses** to vendors for extending prior security analyses of their products onto their new, modified instances - whether patched, updated or substantially changed in a form of new versions. When a patch or update is coded, we analyze its code and impact on the product's security.

16. Will you keep the discovered security issues confidential once we've fixed them?

All project information in a commercial security analysis is considered strictly confidential and will never leave our company. This includes all vulnerability information, non-public product information and all other project-related information.

Moreover, once we've set up a commercial agreement with a customer for one of their products and we happen to find a security problem in another product of theirs, we treat that information as if it was covered by the commercial agreement: we analyze the issue, document it and report it in confidence to the vendor. We never disclose it to anyone else.

Our publicly available reports (http://www.acrossecurity.com/advisories.htm) are the result of some of our analyses outside any commercial agreements.

17. How would the process of engaging you in a security analysis look like?

Based on your initial description of your application, any product documentation you can provide and (ideally) our interaction with a live system – production or testing, remote or local in our lab –, we'll make an assessment of the amount of effort needed for a comprehensive security analysis, which would exhaust all of our extensive knowledge of vulnerabilities and attack methods that we've been continually collecting and updating for well over a decade.

In case your budget doesn't allow for investing in the proposed amount of effort, or you have other reasons for reducing the invested effort, we'll locate and mutually decide to exclude certain attacks, vulnerability types and/or certain parts of your product from the analysis while striving to minimize the decrease in expected results. For instance, it is a frequent request from our customers to limit our scope to "OWASP top 10" or "CWE/SANS top 25" vulnerability categories.

When an agreement is reached on the scope of our engagement, we'll prepare a draft Statement of Work (SOW) for your review. Typically, after a few iterations we'll arrive to the final agreement and have it signed.

Before any project information is exchanged, we'll establish a secure communication channel (usually S/MIME or PGP-protected e-mail). If your application is an installable product (e.g., Windows, Linux or mobile application), we'll install it on our own hardware; in case it is on an appliance, we'll find a way to transport it to our lab with minimum cost and effort; finally, if it is a server application (e.g., a web application or an API server) that cannot be installed locally in our lab, you'll provide us with several accounts on a testing or production site.

If so agreed, you will provide the source code and development documentation over an encrypted channel, and we'll begin with our analysis.

Every time we'll discover a security problem, we'll immediately document it and report it to you, providing help with reproducing the issue if needed.

At the end of the project, we'll compile a final project report and present the findings to your executives and engineers in two presentations focused on their respective interests and points of view.

18. What is a "black box" security analysis?

A "black box" analysis is a security analysis of a product without any direct access to the product's source code, development documentation or any other information that isn't publicly available. This is the kind of analysis that can be expected to be performed by independent security researchers, hackers, competitors or your customers.

It includes activities such as monitoring network traffic, USB or serial communication, file usage, registry usage, API calls, DCOM communication, observing running processes, memory consumption, system handles, CPU usage, analysis of communication protocols, encoding algorithms, encryption algorithms, fault injection, configuration analysis and reverse engineering.

The upside of this type of analysis is that you don't need to provide the source code or any non-public information to the analysts. On the downside, however, this can severely limit the results you get: fewer discovered vulnerabilities, lower quality analysis and sub-optimal recommendations.

19. What is a "white box" security analysis?

A "white box" analysis is an extension of, rather than an alternative to, a "black box" analysis. It includes all "black box" activities, and complements them with an analysis of development documents and the source code.

This type of analysis yields maximum results: the highest vulnerability count, best analysis of the discovered issues, and highest quality recommendations.

20. What is a "grey box" security analysis?

A "grey box" security analysis is a mixture of a black box and a white box analysis, i.e., only partial source code and/or development documentation is provided to us by the customer.

21. Do you practice reverse engineering?

We're well experienced in reverse engineering, but we only perform it if the customer expressly authorizes us to do so. Reverse engineering can be very time consuming, but is a good display of what an attacker could do without the source code. Some customers don't provide us with their source code at first, but change their minds in subsequent analyses when we show them that we were able to recreate the source code of some relevant parts of the product. It is then not a difficult decision to provide the source code in future and allow us to analyze it rather than investing their money in our code reconstruction efforts. We strictly follow local and international laws on reverse engineering.

22. How do you protect your customers' sensitive information?

We begin with setting up an encrypted communication channel with the customer, preferably PGP or S/MIME based e-mail, which we use throughout the project for exchanging any information with the customer. *All* project data remains physically inside our company at all times and is stored on encrypted disks.

Furthermore, all project data is only accessible to participating project members. When we need help from our experts outside the project team, we provide information to them on a "need to know" basis. To formally support our efforts in protecting the customers' information, all participating experts routinely sign project non-disclosure agreements before obtaining access to any project information.

We treat all project information, including any source code and vulnerability information, as strictly confidential for an unlimited amount of time, and destroy or return any customer-provided information to the customer upon the project completion.

23. Why should we choose your company?

ACROS Security has been providing application security analyses to leading software vendors, financial institutions, governments and high-tech industry since 1998.

- We have **talented security experts** trained at "thinking like attackers", which is one of the most important skills a successful security analyst must have. And it is this very skill, combined with our constantly updated knowledge of security technologies, vulnerabilities and attack techniques that makes us so efficient in discovering generic and specific vulnerabilities in various kinds of products.
- Our ethical **penetration testing services keep us sharp** and up to date with practical attack techniques and methods, and complement our lab-based analyses in a unique and highly efficient way.
- We occasionally publish (http://www.acrossecurity.com/advisories.htm) security advisories about vulnerabilities discovered in the software we're using ourselves or meet in our customers' systems during ethical hacking engagements. These advisories are only published after the vendor has made a fix available to the users, and their occasional nature is a result of the limited time our commercial engagements allow us to invest in this pro bono work.
- Our analyses and deliverables are of the highest quality available on the market. We find **one critical vulnerability every 2-3 researcher days** on average in a security mature product, and make inexpensive, efficient recommendations and business-oriented assessments and statistics to help managers prioritize their actions.
- We produce **no false positives**, meaning that your engineers will not have to sift through our reports looking for actual vulnerability information. Instead, they'll be able to focus on confirmed problems and their solutions.
- Our results are presented to executives and engineers differently, based on their respective points of interest.
- We generate most of our revenue in the U.S. market, providing our services to some of the most critical and demanding organizations. Large security solutions providers regularly hire our experts to perform security analyses on their products, which protect the ultimately critical information in banks, Fortune 1000 companies, governments, military and intelligence agencies throughout the world.

We firmly believe our services to be among the very best in the global market. Our customers' feedback regularly confirms this belief, and we'll gladly take every opportunity to prove it to you.

ACROS Security: Finding Your Digital Vulnerabilities Before Others Do.

Call +386 2 3000 280 for more information or send e-mail to security@acrossecurity.com.

www.acrossecurity.com