Keeping Software Safe

ソフトウェアを安全に保つ

Sandhya Gupta - February 6, 2019
サンドヒャ　グプタ　2019 年 2 月 6 日

Sandhya is a program manager in California.

サンドヒャは、カリフォルニア在住のプログラムマネージャ

Everyone in the software industry should be aware of the importance of building safe software. Creating a product means not only addressing customer needs, but also security. These essential requirements should not be an afterthought; they must be built into the product.

ソフトウェア業界の誰もが安全なソフトウェアを開発する事の重要性を認識する必要がある。製品を作ることは、顧客の要望を満たすことだけでなく、セキュリティに対処することでもある。これらの必須要件は、決して後付けであってはならず、製品に組み込まれていなければならない。

But even when engineering teams follow best practices and work on incorporating security requirements into the product, companies still see an influx of reported issues.

しかしながら、開発チームがベストプラクティスに則り、セキュリティ要件を製品に組み込んでいても、企業は数多くの問題報告を受けることがある。

Internally reported issues

問題が内部組織によって報告される場合

1. The product core team is looking into an issue and discovered a security bug.
   製品コアチームが問題を調査し、セキュリティ上のバグを発見する
2. Companies often use software internally before rolling it out ("dogfooding"). An issue may be discovered during this time. There are several use cases and personas on how the product can be used. It's impossible to cover everything in testing, so when the product is used across various groups, people have different approaches. Internal employees across different teams may use different personas and execute different code paths. This mimics "real life" testing and helps identify issues that can be addressed before the product is out in the market.
   企業は、正式公開前に社内でそのソフトウェアを使用することがよくある（試験運用）。そのような試験運用期間中に問題が発見されるケースがある。その製品がどのように使用されるかについての事例やペルソナはいくつか存在し、実装試験時にその全てをカバーすることは現実的には不可能である。様々なグループが製品を使用する場合、異なるアプローチをとることになるため、社内の違う組織から構成される社員

は、それぞれ異なるペルソナを用い、異なるコードパスを実行する事になる。これにより「現実世界」での試験を疑似的に行うことができ、製品が市場に公開される前に問題を特定することに役立つ。

3. There are various security tools that do static code scans. Most of these tools need to be configured and provide a list of issues that need to be sorted for false positives.
   静的コードスキャンを行う多種多様なセキュリティツールが存在する。これらのうちの多くは、適切に設定を行う必要があり、そこから誤検出を選別する必要がある問題のリストを供給する。
4. Big companies have internal groups that are dedicated to doing penetration testing. They use homegrown tools by security groups, or tools that are purchased for it. Smaller companies might also employ the services of outside vendors to complete this testing and furnish a report. All issues identified by this report are then further scored on severity and impact.
   大企業には、侵入テストを実施する専門の内部組織があり、セキュリティチームによる自社製のツールや、外部調達したツールを用いる。比較的小規模の企業の場合は、この種の試験を実施し、報告書を提供する外部ベンダのサービスを利用することもある。この報告書によって特定された全ての問題は、その後重大度や影響度に応じてスコアリングされる。

Externally reported issues

問題が外部組織によって報告される場合

These are issues that are identified by sources that are not considered company employees. They can come from partners, customers or anyone who is using the product. All these issues should be carefully evaluated, reproduced, scored and fixed. There should be a mechanism in place for how the fix will be distributed to consumers. Teams are involved in not only fixing it, but also documenting the issue and its effects.

自社社員とはみなされない外部組織によって問題が特定される場合がある。これらの問題は、パートナーや顧客、または製品を使用するあらゆる人によってもたらされる。提供された全ての問題は、慎重に評価され、再現され、スコアリングされ、そして修正される必要がある。また、修正プログラムがどのように顧客に配布されるかについての仕組みが実装されている必要がある。チームは、問題を修正する事だけでなく、対象となる問題とその影響についての文書化にも関わる。

Backport

古いバージョンへの移植

Software can have many versions. Once a security issue is identified, the company may have to backport the fix to other supported versions. This is a very tedious process. The issue needs to be reproduced for all supported versions, then fixed and tested to ensure that it’s correct in each version (and that there are no regressions). Ideally, this test case should be added to the existing test plan; an automated test should preferably be created.

ソフトウェアは、数多くのバージョンをもつ場合がある。セキュリティ上の問題が特定されると、企業はその修正プログラムを、サポートしている他のバージョンにも適用する必要がある。これは非常に退屈なプロセスではあるが、当該の問題は、サポート対象のすべてのバージョンについて再現された上で、修正され、各バージョンについて是正されている（加えて、修正に伴う副作用で生じた不具合が無い）ことを確認するために試験を行う必要がある。理想的には、このようなテストケースは既存の試験計画に組み入れられ、自動的に試験されることが望ましい。

Support/TAM

サポート/テクニカルアカウントマネージャ(TAM)

Any teams working directly with the customer should also be made aware of the issue and its fix. There should be a clear understanding of the details of the security issue, the fix that was released and the core use cases that were affected. Support should proactively identify customers that were impacted and do outreach. They should ensure that the issue is documented and that customers are aware of the issue and the fix.

顧客に対して直接働きかける全てのチームは、対象となる問題とその修正について認識する必ようがある。セキュリティ上の問題の詳細や公開された修正プログラム、また影響をうける主な使用例について明確に理解しておく必要がある。サポート部門は、影響を受ける顧客を先に見越して特定し、支援を行うべきである。彼らは、対象となる問題が文書化され、顧客がその問題と修正プログラムについて認識していることを確認する必要がある。

Technical account managers work with customers who have paid extra for the services (or with big accounts that warrant a single point of contact). These teams will also work to ensure that the fix is available to the customer and that it is applied. They also look into any specific concerns that the customer may have. If the customer has done any customization, they work with them to ensure that the fix does not break custom code.

テクニカルアカウントマネージャ(TAM)は、対象となるサービスに追加費用を支払う顧客（もしくは、単一の窓口対応を保証するような大規模顧客）と共に、問題解決にあたる。これらのチームは、その顧客に対して修正プログラムが利用可能であり、またそれらが適用されているかを確認するために機能する。さらに、TAM は対象となる顧客が持つ可能性のあるあらゆる懸念についても検討を行う。もしその顧客がカスタマイズを行っている場合は、顧客と共に修正プログラムがカスタマイズしたコードを壊さないかどうかを確認する。

Documentation

文書化

A knowledge base article should be created. It should list details, be self-explanatory and reference the release/hotfix/patch.

文書化された知識ベースが構成される必要がある。詳細が一覧表示され、一目瞭然であり、対象となるリリースや緊急修正モジュール、パッチプログラムが参照可能となっている必要がある。

Community

コミュニティ

If there are any public-facing platforms or communities created, a blog should be created, followed by answers to questions posted.

公開されたプラットフォームやコミュニティが存在する場合、質疑応答が投稿可能なブログを実装するべきである。