The Anatomy of a Security Advisory

Since 2002, Flexera's Secunia Research team has been delivering security advisories that provide reliable, curated, actionable vulnerability intelligence.

What is a Security Advisory?

A security advisory is a summary of the work that Secunia Research performs to communicate standardized, validated and enriched vulnerability research on a specific software product version.

We issue Secunia Research criticality ratings and common vulnerability scoring system (CVSS) metrics after a distinct analysis in the advisories. This dual rating method allows for a much-improved means of prioritizing by criticality—delivering a review that includes product context and related security best practices.

A *rejection advisory* issued by the research team issues means we've determined it's not worthy of your attention. This advisory comes if a vendor issues an advisory acknowledging vulnerabilities that we don't believe to be valid—and would have a product solution we aren't recommending or exceeding already. We send that out to save you considerable time.

If someone other than the vendor issues an advisory and we don't believe to be valid, we discard it. We take that action so you don't waste your time processing inconsequential vulnerability information.

Elements of a Secunia Advisory

- **Secunia Advisory ID (SAID)**–the unique, identifying number for the Secunia Advisory or rejection advisory
- **Creation date**–the date the advisory was created
- **Modification date**–the date the advisory was last edited
- **Criticality**–a five-level criticality identifier

  The Secunia Research criticality level is useful because CVSS scores tend to use fewer of the lower-end CVSS scores. This imbalance results in a top-heavy scale. You can be assured that the whole spectrum of the range is used with the Secunia Research's criticality rating. This rating system ensures prioritization by precision.

  We set the criticality of a Secunia Advisory to the highest level of each enclosed vulnerability. Generally, the criticality level of a vulnerability can be modified if there are limitations concerning who can exploit it and if exploitation becomes increasingly difficult. Examples include when authorization is required for exploitation or when certain network restrictions apply, like when local area network access is required. Additionally, vulnerabilities where the attack complexity is increased (e.g., if a man-in-the-middle position is required) typically experience a decrease in their criticality level.

The possible values are:

- **Extremely critical**
    - This value is typically used for remotely and easily exploitable vulnerabilities that are otherwise designated “highly critical” but also have been exploited in the wild before their publication (zero-day). These vulnerabilities typically exist in services like FTP, HTTP and SMTP or specific client systems such as email programs or browsers. Operating systems can also be prone to them—e.g., when font handling is performed on operating system level.
- **Highly critical**
    - This value is generally used for remotely and easily exploitable vulnerabilities that can lead to system compromise.
    - Successful exploitation doesn’t usually require any interaction, but there are no known exploits available at the time of disclosure.
    - These vulnerabilities typically exist in services like FTP, HTTP and SMTP or specific client systems such as email programs or browsers. Operating systems can also be prone to them—e.g., when font handling is performed on operating system level.
- **Moderately critical**
    - This value is usually used for remotely and easily exploitable denial-of-service vulnerabilities against services like FTP, HTTP and SMTP. Additionally, easily exploitable vulnerabilities that could lead to information disclosure or affect the integrity of a product can result in this criticality level.
- **Less critical**
    - This value is typically used for cross-site scripting and local privilege escalation vulnerabilities.
- **Not critical**
    - This value is generally used for local denial-of-service vulnerabilities but also those with extremely limited impact like URL redirection vulnerabilities.
    - **Zero Day**—a “yes” or “no” is given to indicate if this is a zero-day vulnerability. A zero-day vulnerability is a vulnerability that’s been exploited prior to its disclosure. To ensure that such zero-days are properly prioritized, their criticality is increased to the next criticality level.

- **Impact**
    - **Brute force**–used in cases where an application or an algorithm allows an attacker to guess passwords easily.
    - **Cross-site scripting**–cross-site scripting vulnerabilities allow a third party to manipulate a web application’s content or behavior in a user's browser, often without compromising the underlying system. Different cross-site scripting vulnerabilities are also classified under this category, including script insertion (also called persistent cross-site scripting) and cross-site request forgery. The impacts of cross-site request forgery can vary depending on the actions that can be triggered. Cross-site scripting vulnerabilities are often used against specific users of a website to steal their credentials, conduct spoofing attacks or trigger actions on behalf of the user.
    - **DoS (denial of service)**–this includes vulnerabilities ranging from excessive resource consumption (e.g., causing a system to use a lot of memory) to crashing an application or an entire system.

- **Exposure of sensitive information**–vulnerabilities where documents or credentials are leaked or can be revealed either locally or remotely.
- **Exposure of system information**–vulnerabilities where excessive information about the system (e.g., version numbers, running services, installation paths and similar) are exposed and can be revealed remotely and, in some cases, locally.
- **Hijacking**–covers vulnerabilities where a user session or a communication channel can be taken over by other users or remote attackers.
- **Manipulation of data**–this includes vulnerabilities where a user or a remote attacker can manipulate local data on a system but not necessarily be able to gain escalated privileges or system access. The most frequent type of vulnerabilities with this impact are SQL-injection vulnerabilities, where a malicious user or person can manipulate SQL queries.
- **Privilege escalation**–covers vulnerabilities where a local user can conduct certain tasks with the privileges of other users or administrative users. This typically includes cases where a local user on a client or server system can gain access to the administrator or root account, taking full control of the system.
- **Security bypass**–covers vulnerabilities or security issues where malicious users or people can bypass specific security mechanisms of the application. The actual impact varies significantly depending on the design and purpose of the affected application.
- **Spoofing**–covers various vulnerabilities where malicious users or people can impersonate other users or systems.
- **System access**–covers vulnerabilities where malicious actors can gain system access and execute arbitrary code with the privileges of a local user or the underlying system.
- **Unknown**–covers various weaknesses, security issues and vulnerabilities not covered by the other impact types or where the impact is not known due to insufficient information from vendors and researchers.

- **Where (attack vector)**
  - **Local system**
    - Local system describes vulnerabilities where the attack vector requires that the attacker be a local user.
  - **Local network**
    - From local network describes vulnerabilities where the attack vector requires that an attacker is situated on the same network as a vulnerable system but not necessarily a LAN (e.g., physically adjacent Bluetooth).
    - Additionally, this also applies to the internal network layer separating host and guests when using virtual machines.
    - This category covers vulnerabilities in certain services (e.g., DHCP, RPC and administrative services) which shouldn't be accessible from the Internet—but only from a local network and optionally a restricted set of external systems.
  - **Remote**
    - From remote describes vulnerabilities where the attack vector does not require access to the system or a local network.
    - This category covers services that are acceptable to expose to the Internet (e.g., HTTP, HTTPS and SMTP) and client applications used on the Internet and specific vulnerabilities—where it's reasonable to assume that a security-conscious user can be tricked into performing certain actions.

- **Solution status**–if a fix and what kind is available. Possible values:
  - **No**–reserved for rejection advisories where an action is not required.
  - **Vendor patched**–where a direct solution can be applied for the specific product and version branch. For example, this could mean applying a hotfix and a minor version update.
  - **Partial fix**–this solution status is used if not all products or not all vulnerabilities covered in the Secunia Advisory have a direct solution concerning the specific product and version branch.
  - **No fix**–no direct applicable solution for the product and version branch applicable to the Secunia Advisory exists. This could still mean that an upgrade path may still be available.
  - **Vendor workaround**–used when manually actions are required either to resolve (e.g., application of GIT commits) or to mitigate the vulnerabilities within the Secunia Advisory (e.g., application of GIT commits or altering of system configuration). To note, if such a manual change would result in a loss of functionality when using a product, such manual changes are typically not considered valid.
- **Secunia CVSS scores**–following our consistent process executed by Secunia Research we leverage the CVSS standard to identify a single CVSS score for a Secunia Advisory. As Secunia Advisories may bundle multiple vulnerabilities, the vulnerability with the highest CVSS score dominates the CVSS score for the Secunia Advisory.
- **CVE references**–CVE (common vulnerabilities and exposures) identifiers represent a unique, standardized identification for a given vulnerability or exposure. Searching on a CVE reference (e.g., CVE-2009-3793 or simply 2009-3793) will find all Secunia Advisories in the database that list that CVE as a reference. An advisory can contain more than one CVE reference. As not every valid vulnerability is assigned a CVE identifier, not every Secunia Advisory may feature an associated CVE reference(s). Furthermore, as the CVE identifier assignment is frequently performed without detailed analysis, not all CVE identifiers necessarily represent valid vulnerabilities.
- **Threat score**–a 0 to 99 value where the higher the value, the more likely the affected operating system and software is exploited.
- **Software**–the software title in question.
- **Common platform enumeration (CPE)**–when available a CPE and a link to that CPE is provided.
- Advisory details:
  - **Description**–an easy to understand, explanatory statement regarding applicable vulnerabilities
  - **Solution**–what can be done to mitigate the vulnerabilities, such as to which version one should update or upgrade
  - **Provided and/or discovered by**–credit for the disclosure(s)
  - **Changelog**–a quick record of changes to the advisory