Arista Vulnerability Management Process

Vulnerability Detection. Arista uses the following tools to detect vulnerabilities.

- CVE Scanner is an Arista automated process that searches for publicly disclosed vulnerabilities in the open source packages used in EOS® and CloudVision® Portal. It works by automatically downloading the database of known issues from the National Vulnerability Database (https://nvd.nist.gov/) and then cross-checking the version for each vulnerability against the versions of RPMs in all releases that have not yet reached end-of-life. Upon identifying a match for a vulnerability, a new bug is automatically filed and an engineer will investigate the potential problem. This automated process provides Arista with the ability to quickly correlate potential security vulnerabilities in the public domain.
- Arista also keeps open lines of communication with its third-party vendors that provide software and hardware solutions to Arista products. In the event that a security issue is found with the 3rd party vendors product they are encouraged to reach out to Arista's PSIRT team and discuss the issue. Arista will treat these issues in the same manner as any other security issue discovered.
- Arista PSIRT engineers conduct ongoing, detailed, security reviews of the Arista written code base to check for potential vulnerabilities and issues. Special attention is paid to areas of code believed to be at higher risk, such as those that parse user input or handle external packets.
- Any potential security issue identified is reviewed for severity and potential impact. As part of this internal issues are scored using CVSSv2 and CVSSv3 scoring. If the usage of a publicly disclosed piece of vulnerable software differs significantly from the original reported usage, Arista may re-report the score with regards to how vulnerable Arista products are.

Vulnerability Communication

- If the issue impacts Arista products, an appropriate solution or set of solutions to the problem will be provided. The solutions can include a recommended configuration change, software patch, new software image, or other procedures that are appropriate to mitigate or fix the vulnerability.
- Details of the security vulnerability and associated solutions are then documented publicly via a security advisory on the Arista website: https://www.arista.com/en/support/advisories-notices

Security Assessment Testing

- Arista performs regular internal security assessment testing on our software products. These internal tests are done for every major software release (multiple releases per year). An example of the internal security test cases is included below in Appendix A.
- The findings of these tests are reviewed to find ways in which to improve or harden the EOS software.

Safe Choices in Design and Runtime

- Much of the code in Arista product is written in a meta-language that combines the speed of C++ with the safety checks that one would expect to see in a more secure language. As a result, many common types of common programming issues are caught during development or not able to occur in the framework that is used. This includes, but is not limited to:
    - Prevention of resource leaks via Valgrind and reference counting.
    - Pointers and data structures are initialized to safe, NULL, values upon creation.
    - Common memory errors such as buffer overflows are prevented by doing smart bounds checking on each operation.
    - Concurrency issues (i.e. race conditions) are prevented by treating each program as a single thread of execution which communicates with other programs in order to share data.
- Both the design of new Arista features and maintenance of existing features are done with security as a goal.
    - Engineers are provided training on secure coding practices and how to implement them in their code. By having a series of guidelines and examples engineers can create features that are designed to be secure from the start and can recognize previously written insecure designs.
    - The usage of security critical open source libraries is limited to a few well understood libraries. This serves to limit the surface of attack as well as make analyzing the usage of said libraries in the codebase easier.
    - Awareness and review of common attack vectors and the associated mitigations is an important part of security at Arista. The PSIRT team makes sure to stay aware of common patterns in insecure code and how to detect them. Information on rising trends is integrated into the training as well as company wide announcements. By making sure to keep a dialogue open within the company on security, engineers on all teams are able to keep secure coding principles in mind when writing code.

Vulnerability Mitigation on Running Systems

- In the event of a vulnerability that affects Arista products, Arista is oftentimes able to provide a hotfix to mitigate the issue. This is an extension that can be installed on a running system and will fix the problem with a minimum of downtime. While not all fixes are available as a hotfix, here is an example of how this hotfix scenario could look:
    - The SSH Server is found to be vulnerable to a publicly disclosed CVE.
    - Arista creates a hotfix to resolve the problem.
    - The hotfix is installed on a switch, the SSH server goes down for approximately 1 second while it is restarted. No other services on the switch are affected.
    - The fix is in place and can persist across reboots of the switch until a newer image with the fix integrated can be loaded.

Appendix A: Arista's internal security assessment test cases include, but are not limited to the following:

- External host fingerprinting for display of compromising information.
- Automated vulnerability scanning for checking installed software against known CVEs.
- Validation of automated scanning results by an Arista engineer who specializes in security.
- Attempt Proof-of-Concept exploitation against running host, for vulnerabilities with documented ability to do so.
- Internal host configuration review including the boot loader, kernel (hardware drivers, modules, patches, etc), ipv4 and ipv6 stack and other services running on the host.
- Use of open-source tools such as nmap and gcov to ensure thorough coverage and understanding of the code deployed.
- Testing to standards defined by the DoD DISA STIG configuration and best practices. Arista uses the DoD DISA standards as our baseline for secure computing since they provide a high level for initial entry and cover threats one would expect to find in a datacenter.
- Review of EOS and CloudVision Portal software architecture and design documentation, with a focus on external attack vectors, to identify design flaws from a security perspective.
- Regular review of best practices for securing and hardening Arista EOS and CVP to ensure the security of the network. Best practices maintained in the Arista EOS Hardening Guide, which is a living document stored here: https://eos.arista.com/arista-eos-hardening-guide/

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062