

Security

1. Is it safe to run applets written in the Curl language? Aren't applets a security problem?

Curl® applets are safe. The Curl Runtime Environment makes conservative security choices to protect users, computers, and networks from maliciously written Curl applets. When you run a Curl applet, your computer downloads the applet in an easy-to-analyze form, just the way the applet's author wrote it. The Curl Runtime Environment can easily analyze a Curl applet to see what the applet is trying to accomplish. It refuses to do dangerous things that a hostile applet might want to do.

2. Can hackers use the Curl® language to write viruses and Trojan Horses? What protects my local files and local network from hostile Curl applets?

The Curl Runtime Environment carefully guards the end-user's computer and local network from malicious applets, so as to close the virus-writer's toolbox. For example, the Curl Runtime Environment will not let a Curl® applet examine or alter the end-user's files without explicit permission from the user. Similarly, when an applet wants to access the Web or make any sort of network connection, the Curl Runtime Environment will only allow access to specific servers. These servers have explicitly been configured to say that they allow Curl applets from a given server to access them. Finally, the Curl Runtime Environment will not let an unprivileged applet execute ActiveX® controls, scripts, or macros written in other languages. With these controls in place, a would-be virus cannot infect the user's disk, cannot spread itself across the user's network, and cannot access private data.

3. Security rules are so cumbersome. Is Curl security hard to use?

We've designed the Curl® security features to be easy-to-use for content authors, system-administrators, and end-users:

- For end-users, Curl security is mostly invisible, with very little security-related user-interface. There are no sandbox access-rules, and few security-related dialogs or windows. Also, nothing unsafe is allowed until the user explicitly allows it.
- For server system-administrators, Curl security is easy to set up and requires little maintenance. Even for large server deployments and large user-populations, Curl security configurations are easy to write, easy to read, and take up less than one kilobyte of text.
- For content-authors, Curl security avoids the complexity of Java's "signed, trusted applets." A normal Curl applet can safely do some local I/O and allowed network access without being privileged. However, it isn't powerful enough to harm the end-user's resources or leak information across a firewall.

4. How does Curl security compare to Java's?

The security features of the Curl® language are similar to Java™'s: strong typing, simple name-scoping, automatic memory-management, and a sandbox to control access to host and network resources. One big security difference is in the way the Curl Runtime Environment analyzes and runs applets.

In Java, the applet-author compiles applets into bytecode, then the user's JVM runs the applets by interpretation or native-code compilation. Before the JVM runs the applet, though, the JVM analyzes the applet's bytecode for security problems. The JVM uses complex mechanisms for bytecode verification and class loading, so as to ensure that the applet obeys Java's type-safe language security rules. However, a hostile author can write his applets in Java bytecode directly, in hope of bypassing Java's security mechanisms. Researchers and hackers have found many bugs and security holes in these complex JVM mechanisms.

A Curl applet or package is received by the Curl Runtime Environment as either source code or as a .pcurl file. A .pcurl applet is a non-text version of Curl source-code which retains all the data-type information that was in the original, human-readable Curl applet. Because both kinds of Curl code retain their variables' data-types, the Curl compiler can always control very carefully the process of translating typed source code to machine code. By retaining all data-types in every applet, the Curl technology greatly simplifies the process of ensuring correct behavior of the code. Thus, your Curl Runtime Environment can easily, reliably, and fully analyze each applet you download for security.

Another difference is that the Curl Runtime Environment runs each Curl applet in a separate Curl process, so it can easily keep different applets' activities and resources separate. With Curl processes, the user can terminate applets, and Curl can properly reclaim any resources the applets were using. In contrast, Java's JVM runs Java applets as threads which share the JVM's resources in one big pool. We believe Curl applet processes are more robust than Java's threads.

The Curl Runtime Environment also imposes resource consumption limits on Curl applets. Therefore, one applet cannot use up so many resources that it interferes with other Curl applets or other programs on your computer.

Finally, the Curl Runtime Environment allows access to network resources like web sites and other network servers only if those servers have been explicitly configured to allow those accesses. This behavior allows a more general use of network resources than Java (which only allows access to things from the server that the Java applet was loaded from). At the same time, the Curl approach never allows access to anything that the administrator of that server hasn't explicitly authorized.

5. Can I sign Curl applets?

Yes, the Curl platform includes support for signed applets.

6. Under what circumstances can a Curl applet touch the user's local files?

The Curl Runtime Environment does not let unprivileged Curl® applets touch local files unless the user gives explicit permission through a special dialog that warns the user about the security implications. The interaction is structured so that the the first time a file is accessed, the user must navigate to (or type, or paste) the file name to be opened by the applet. This insures that an inadvertent user click will not open a file.

The Curl API also includes the procedure request-local-data-permission, which will ask the user to grant an unprivileged applet permission to store an unlimited amount of local data in a directory which will be created by the Curl Runtime Environment.

A user can also give full privilege to an applet. Privileged applets can touch any file that the user is allowed to touch, once the end-user has explicitly granted privilege to the applet in the Curl Control Panel.

7. What if a system administrator does not want end users to grant privilege to applets?

The Curl Runtime Environment consults a configuration file before allowing users to control applet privileges. The runtime environment can be configured to disable the control panel that is used to grant privilege to applets. The configuration file can also be used by a system administrator to completely restrict access to all applets, only allowing access to applets served from specified URLs.