

ACCESS™ White Paper

# ACCESS Linux Platform™ Security Policy Framework

# ACCESS Linux Platform™ Security Policy Framework

## **ACCESS' open framework ensures a flexible security policy for today's smartphones**

*As mobile phones have grown more sophisticated, packing more processing power, memory, and sophisticated system software and offering users more functionality and the possibility to add applications and download content, they have also become more interesting targets for malicious attacks.*

Most basic cell phones are closed systems and lack the system resources to accommodate new applications, including those that can cause harm. So traditionally, security has not been an issue. More advanced classes of phones that will only permit applications to be created in heavily controlled environments, such as Java™, are less secure than closed systems, but generally secure enough to withstand serious attack. Security becomes a major problem with the most sophisticated mobile devices, smartphones, which allow users and others to add new applications and content from a variety of sources. Applications gain access to phone resources via an operating system API, thereby introducing concerns over application behavior.

Badly behaved applications, or “malware,” can cause harm ranging from annoyance to much worse: changing icons, making unauthorized international calls, connecting to remote servers, accessing private user or application data, and rendering the phone completely inoperable. Malware may be unintentionally downloaded to the phone by the end user or propagated from another phone

via Bluetooth. In some situations, the attacker is actually the device owner, using hacking tools to try to access premium services for free.

A mobile phone security framework requires flexibility. Network operators, device manufacturers, as well as IT departments and end users may each want to customize the phone by adding applications, with each party making certain that the applications they add are well-behaved. What is less certain at the outset is what enforcement role each party should play. A one-size-fits-all static security model may prevent malware from being added to the device, but it will also prevent useful applications from being added.

### **The key: flexible sandboxing**

The ACCESS Linux Platform's Security Policy Framework (SPF) achieves flexibility through a technique called “flexible sandboxing,” which adds much latitude in determining when and how applications can access system resources.

“Sandboxing” conventionally refers to the practice of enforcing security through an API and resources policy which governs and limits application behavior on the device. With conventional sandboxing, every third-party application is given the same set of privileges, no matter where it came from or what testing it has passed undergone. This all-or-nothing approach means that no one but the network operator can write, for example say, an application that does instant messaging-because that application could not open a network connection. This is the case even if the network operator wishes otherwise.

The ACCESS Security Policy Framework adds flexibility to the sandboxing model by allowing network operators and handset manufacturers to define and enforce their own security policy. Each party in this value chain can also defer some decisions to a stakeholder closer to the consumer.

The SPF defines not one, but a series of sandboxes, each with an independent set of privileges. With this approach, device manufacturers and network operators specify their own policy that determines which applications are granted which privileges. Rather than forcing all developers to adapt to a “one size fits all” security approach, the ACCESS SPF allows manufacturers and network operators to match policy enforcement with the degree of trust they have in the code and the code’s developer.

What this means in practice is that the SPF restricts access to data files, APIs and configuration information on a case-by-case basis. Some applications will have access to the phone’s global settings, while others will not. Some applications will run with “root permissions” that give them broad access to data. Others will be far more restricted. In addition, every signed application has access to its own data files, which cannot be accessed by other applications.

With the SPF’s flexible sandboxing, every application, even an unsigned one that has not undergone any third-party testing, might be given some low-level privileges, such as the ability to play a sound or put a window up on the screen. But that same unsigned application would not be allowed to open a network connection or dial a phone number.

The next level of privilege might allow an application to send an SMS message or open a specified set of files. Beyond that, an application might be able to reconfigure Wi-Fi settings or, at the most trusted edge of the security spectrum, might even have

permission to re-flash the ROM, replacing part or all of the phone’s own operating system.

None of this is fixed: service providers have the flexibility to set their own security policies that are as stringent or lenient as they choose. A network operator, for example, could set policy so that a developer could create a signed application that leverages a user’s contact list, while still restricting the behavior of other unsigned applications whose pedigree is less known.

### Working with multiple policies

With the SPF, each stakeholder—including handset manufacturers, system integrators, network operators can define a security policy for their “domain,” thereby creating several security policies on a given device at any given time. As an illustrative example, consider the default behavior, as defined by the security policy, when an application attempts to access a controlled resource like the boot device or SMS subsystem:

Resource	ACCESS	Handset Manufacturer	Network Operator	ISV signed	ISV unsigned
Boot Device	Always	Always	Always	Never	Never
SMS Subsystem	Always	Always	Always	Always	Ask
Network	Always	Always	Always	Always	Ask
Application data	Always	Always	Always	Always	Never
Calendar	Always	Always	Always	Ask	Never
Address Book	Always	Always	Always	Ask	Never
Audio Hardware	Always	Always	Always	Always	Always

Following this decision matrix, software signed by ACCESS, the handset manufacturer, or the network operator may access any of the systems at any time. The independent software vendors’ signed applications have more restrictions; unsigned

applications would be even further restricted. For example, say an application wants to access the SMS subsystem to send a message. If the application is unsigned, the user must first be asked, and must grant permission before any message is sent. But if the application is signed, access to SMS privileges will be automatically granted.

## Application testing and signing

These security policies reside in a settings file, which can be modified by ACCESS and the device manufacturer to reflect the requirements of the network operator. The policies are enforced through the use of standard X.509 certificates provided by a trusted certificate authority. This approach provides maximum flexibility to the network operator and device manufacturers

Third-party developers can get an application signed for the platform by submitting the application to the testing and signing program. If the developer is new to the program, their company identity will be verified. ACCESS's third-party testing house will test the

application for good, predictable, well-documented behavior on the device, working with the developer, if necessary. Finally, the certificate authority signs the application and provides it back to the developer, and the process is complete.

In conclusion, having a solid, yet flexible, security solution is key to enabling an open mobile environment, where the basic functionality of mobile devices can be extended with new applications and new capabilities. Flexible security policy enables the mobile communications industry and its customers to enjoy the benefits and openness of Linux without putting the user's data or the cellular networks at risk. The ACCESS Linux Platform adds a comprehensive security solution appropriate to these kinds of devices to the Linux operating system – yet another example of how ACCESS has enhanced Linux.

For more information on the ACCESS Linux Platform, please visit [www.access-company.com](http://www.access-company.com).



### About ACCESS CO., LTD.

ACCESS CO., LTD., is a global company providing leading technology, software products and platforms for Web browsing, mobile phones, wireless handhelds and other networked devices. ACCESS' product portfolio, including its NetFront(tm) Browser, Garnet(tm) OS (formerly Palm OS(r)) and ACCESS Linux Platform(tm), provides customers with solutions that enable faster time to market, flexibility and customizability. The Company, headquartered in Tokyo, Japan, operates 29 subsidiaries and affiliates within Asia, Europe and the United States. ACCESS is listed on the Tokyo Stock Exchange Mothers Index under the number 4813. For more information about ACCESS, please visit <http://www.access-company.com/>.

Copyright © 2007, ACCESS Systems Americas, Inc. ACCESS, ACCESS Linux Platform and certain other trade names, trademarks and logos are trademarks which may be registered in the United States, France, Germany, Japan, the United Kingdom, and other countries and are owned by ACCESS Systems Americas, Inc. or its affiliates. These marks may not be used in connection with any product or service that does not belong to ACCESS Systems Americas, Inc. or its Affiliates (except as expressly permitted by a license with ACCESS Systems Americas, Inc.), in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits ACCESS Systems Americas, Inc., or its subsidiaries or affiliates. All other brands and trademarks used herein are or may be trademarks of, and are used to identify other products or services of, their respective owners. All rights reserved.

The registered trademark LINUX® is owned by Linus Torvalds, owner of the mark in the U.S. and other countries, and licensed exclusively to the Linux Mark Institute, from whom ACCESS obtained its non-exclusive license to the mark.

Java is a trademark of Sun Microsystems, Inc

Mozilla is a registered trademark of the Mozilla Foundation.