

Java7u11 sebezhetőség

Adam Gowdiak jelezte pénteken a [full disclosure](#) levelez listán, hogy az Oracle által frissen kiadott Java 7u11 verzió – amely javítja az egyik aktívan kihasznált biztonsági hibát – további két biztonsági hibát tartalmaz, amelyekkel ki lehet törni a sandbox-ból, s ezeket a sebezhetőségeket is elkezdték kihasználni.

További probléma, hogy az Oracle csak egy hibát javított a friss verzióban, amint erről Esteban Guillardoy ír egy friss blogbejegyzésben:

405	/**	405	/**
406	* Is this method a caller-sensitive method?	406	* Is this method a caller-sensitive method?
407	* I.e., does it call Reflection.getCallerClass or a similar method	407	* I.e., does it call Reflection.getCallerClass or a similar method
408	* to ask about the identity of its caller?	408	* to ask about the identity of its caller?
409	*/	409	*/
410	// FIXME: Replace this pattern match by an annotation @sun.reflect.CallerSensitive.	410	// FIXME: Replace this pattern match by an annotation @sun.reflect.CallerSensitive.
411	static boolean isCallerSensitive(WebsterName name) {	411	static boolean isCallerSensitive(WebsterName name) {
412	assert(name.isImmutable());	412	assert(name.isImmutable());
413	Class<?> defc = name.getCallerClass();	413	Class<?> defc = name.getCallerClass();
414	switch (name.getScheme()) {	414	switch (name.getScheme()) {
415	case "http://":	415	case "http://":
416	return defc == java.security.AccessController.class;	416	return defc == java.security.AccessController.class;
417	case "geturl":	417	case "geturl":
418	return defc == sun.misc.Unsafe.class;	418	return defc == sun.misc.Unsafe.class;
419	case "lookup":	419	case "lookup":
420	return defc == java.lang.invoke.MethodHandles.class;	420	return defc == java.lang.invoke.MethodHandles.class;
421	case "invoke":	421	case "invoke":
422	return defc == java.lang.reflect.Method.class;	422	return defc == java.lang.reflect.Method.class;
423	case "get":	423	case "get":
424	return defc == java.lang.reflect.Method.class;	424	return defc == java.lang.reflect.Method.class;
425	case "getBoolean":	425	case "getBoolean":
426	return defc == java.lang.reflect.Method.class;	426	return defc == java.lang.reflect.Method.class;
427	case "getByte":	427	case "getByte":
428	return defc == java.lang.reflect.Method.class;	428	return defc == java.lang.reflect.Method.class;
429	case "getChar":	429	case "getChar":
430	return defc == java.lang.reflect.Method.class;	430	return defc == java.lang.reflect.Method.class;
431	case "getDouble":	431	case "getDouble":
432	return defc == java.lang.reflect.Method.class;	432	return defc == java.lang.reflect.Method.class;
433	case "getFloat":	433	case "getFloat":
434	return defc == java.lang.reflect.Method.class;	434	return defc == java.lang.reflect.Method.class;
435	case "getInteger":	435	case "getInteger":
436	return defc == java.lang.reflect.Method.class;	436	return defc == java.lang.reflect.Method.class;
437	case "getLong":	437	case "getLong":
438	return defc == java.lang.reflect.Method.class;	438	return defc == java.lang.reflect.Method.class;
439	case "getShort":	439	case "getShort":
440	return defc == java.lang.reflect.Method.class;	440	return defc == java.lang.reflect.Method.class;
441	case "newInstance":	441	case "newInstance":
442	if (defc == java.lang.reflect.Constructor.class) return true;	442	if (defc == java.lang.reflect.Constructor.class) return true;

<http://immunityproducts.blogspot.com.ar/2013/01/confirmed-java-only-fixed-one-of-two.html>

Célszer a Java applet futtatást csak akkor és csak addig engedélyezni, amikor és ameddig feltétlenül szükséges... a legfrissebb Java verzió nem ad elég védelmet, amíg a sandbox mechanizmus tüzetes átvizsgálása meg nem történik az Oracle részéről...