

IBM Podcast

[MUSIC]

MATHENY: Welcome to this IBM podcast. I'm Angelique Matheny. The title of this podcast is Flash and Flex Security.

Ayal Yogev, Senior Security Researcher with IBM Rational, will introduce Flash and Flex Security, presenting the most prevalent security concerns in these technologies and will suggest techniques on how to test for their existence as well as provide recommendations and best practices for avoiding such security vulnerabilities. Hi, Ayal. Thanks for joining us. Welcome to the podcast.

YOGEV: Hi, thanks.

MATHENY: We've got a lot to cover today, so let's get started. Let's start with this. Could you tell us a little about flash?

YOGEV: Yes. Adobe Flash is the multimedia platform that was introduced in 1996 by Macromedia, which was later acquired by Adobe. Flash is generally used for fading animations and integrating video into Web pages. More recently, Flash was used for developing rich intelligent

applications. It's also often used in advertising for interactive media.

Flash is a basic scripting language called Action Script, which was initially designed for controlling simple two-dimensional vector automations made in Flash. Later versions of Action Script added advanced functionality allowing for the creation of Internet applications with streaming media.

The reason that Flash is so popular is because many Web developers are familiar with Flash and feel more comfortable using it, instead of other technologies such as Java script or AJAX. Flash then itself is very popular and installed over 99 percent of Internet enabled PCs which is actually more than Internet Explorer.

MATHENY: That is very popular. So how can Flash movies be accessed by users?

YOGEV: There are two ways for accessing Flash movies. The first one is to browse to an HTML page containing a Flash movie as an object. The movie will be loaded by the browser inside the HTML page.

Another option is to access the Flash movie directly by browsing to its URL. In this case, depending on the

behavior of the browser, a dummy HTML page containing the Flash movie will be created.

MATHENY: What else do we need to know before moving on to Flash security?

YOGEV: Before moving on to Flash security, we must talk about something called global Flash variables. Like most scripting languages, action script supports global variables that can be accessed from anywhere in the movie. What is unique about Flash is that these global parameters can also be assigned from outside the movie.

A very common use of global Flash variables is actually using the global variable checking to see if it's undefined and if so, setting a different value. The reason this code is so common is that many Flash developers are unexperienced programmers and therefore they tend to copy pieces of code from other movies.

When this piece of code is used, the value of the global variable can be controlled from outside the movie, and this can cause significant security risks.

MATHENY: And how can the value of a global Flash variable be controlled from outside the movie?

YOGEV: There are actually three ways for assigning values to global Flash variables from outside the movie. The first one is direct [director forms]. When you access the movie directly, it is possible to assign values using URL format.

You just add a question mark following the movie's URL, then the name of the variables and their value separated by ampersand.

Another way is when embedding the Flash movie inside an HTML page, [INAUDIBLE] the movie setting the data attribute of the object tag can contain values for the global variables using the URL format with [INAUDIBLE].

The third way is using a special attribute in the HTML object tag called Flash value. We can set the values of the global variables in a similar fashion and again separate the variables by ampersands.

MATHENY: Okay. We've covered some basic Flash concepts. Let's move on to discussing Flash security. What can happen if someone can take control of a global Flash variable?

YOGEV: Well, if someone were able to take control of a global Flash variable, they can trigger numerous attacks

such as call side Flashing, call side scripting through Flash and phishing. They might even be able to change the flow of the Flash movie itself.

MATHENY: What is cross-side Flashing?

YOGEV: Cross-side Flashing occurs when a malicious Flash movie is loaded into a vulnerable movie and is given access to the same sandbox. When this happens, the malicious movie is able to retrieve and change the values of variables inside the vulnerable movie or to access the dome the movie is embedded in, thus shattering the same origin policy. This can be done when a global Flash variable is used inside native action script functions that load other Flash movies such as load movie.

MATHENY: Ayal, you also mentioned cross-side scripting through Flash. Could you explain this as well?

YOGEV: Yes. This is an extremely harmful attack which is basically triggering a classic call side scripting using a vulnerable Flash movie. This can be done when a global Flash variable is used in one of the potentially dangerous native functions that may trigger JavaScript or action script code such as get URL or load functions.

Call side scripting is a very serious security risk that may

be used to craft powerful phishing attacks, [INAUDIBLE] to sensitive data or resulting in user impersonation.

MATHENY: That is very serious. Are there Flash movies that are more vulnerable to these threats than others?

YOGEV: Well, as you've seen, Flash implements security risks like call side Flashing and Flash flow manipulation as well as create a new platform for known security vulnerabilities like call side scripting and phishing.

We must keep in mind that all Flash applications are a possible security threat. Even the simple marketing ads we see on many Web sites or the generated Flash movies created by applications like Adobe Presenter may be vulnerable and compromise the entire site.

MATHENY: Okay. Let's talk about another technology called Flex. Can you explain a little about this technology for us?

YOGEV: Flash is a relatively new technology developed by Adobe. It is used for developing cross platform, rich Internet applications based on Adobe Flash. Recent versions of Flex are based on NXML, which is an XML markup knowledge used to build the graphical user interface, and on Action Script 3, the new and object-oriented version of Action

Script. Flex combined HTML and Flash and communicates with [sub] services using a messaging format called AMF.

MATHENY: So what is AMF?

YOGEV: AMF stands for Action Script Message Format. It is used by the Flash application to interact with the server behind the scenes. This is somewhat similar to how JSON is used in AJAX. The AMS message format is binary and it is usually delivered over HTTP.

It is important to know that server side components or are custom components and thus may suffer from the same security vulnerabilities that any Web service or Web application may suffer from.

MATHENY: Now that we know what Flex is, can you talk about Flex security?

YOGEV: Flex combines Flash movies on the client side and also communicates with a server. This is why Flex is susceptible to both client side and server side attacks. On the server side, attacks can be delivered to the server using AMF.

All regular server side risks apply in this case including but not limited to, SGL injection, buffer overflows and

session fixation. On the client side, all Flash-based client side risk we have seen may apply since the application is based on Flash movies.

MATHENY: Okay. We've covered the basics of Flash and Flex and touched on some important security concerns. Let's talk about techniques on how to test for these security vulnerabilities and perhaps some recommendations and best practices for avoiding them. How can Flash movies be automatically tested for security vulnerabilities?

YOGEV: In order to automatically test Flash movies, we must first identify the global Flash variables that can be controlled from outside the movie.

We can find these variables from parameters or the Flash attribute in the HTML page the Flash movie is embedded in, or by identifying the un instantiated variable in the action script code. We then want to understand what these variables are used in potential dangerous Action Script code where these variables are passed as parameters to functions like get URL and load functions.

We also need to remember the sequence of actions leading to that code. For example, we want to remember the sequence of buttons we have to click in order to execute the get URL function in which a specific global function is used. We

then inject values to the global values testing for the vulnerabilities and reload the Flash movie.

In order to validate the vulnerability, we can play the relevant sequence leading to the potential dangerous code and verify whether the attacks succeeded.

We can verify the results by testing browser events, like the appearance of an alert box or validate on the action script level that the functions were in fact caused and executed.

It is also very important to test the movie within its original HTML environment. This is because the Flash movie can interact and change its flow according to the HTML dome during execution. In case the Flash movie is tested as a standalone movie, some pieces of code may not be reachable and thus not testable.

MATHENY: And, what about Flex? Can Flex be automatically tested as well?

YOGEV: Yes. Flex can also be automatically tested. In order to test Flex security, we must test the client side Flash movies and also test the server side code for the AMF messages. During the explore phase, the AMF messages are passed in order to extract the internal structure. Later,

during the test phase, we apply existing payloads to parameters inside the messengers in order to test for security vulnerabilities, like SGL injection or buffer overload.

MATHENY: Ayal, should anything else be tested?

YOGEV: Adobe provides guidelines for coding secure Flash applications. We should also test for Flash best practices and misconfigurations of the Flash environment. For example, the cross domain dot XML policy file is a simple XML file that gives the Flash player permission to access data from a given domain, without displaying the security dialogue.

An over permissive cross domain dot XML file may result in security violations. Another example is giving permissive permissions to the Flash sandbox, allowing the Flash movie and other movies loaded into it to open socket and communicate with the page dome.

MATHENY: Do you have any recommendations for coding secure Flash and flex applications yourself?

YOGEV: The most important recommendation is to validate the values of variables sent to potentially dangerous native functions. Furthermore, it is best not to

use global Flash variables as arguments for these functions for this enables many Flash security vulnerabilities as we have seen.

We also recommend applying minimal permissions in the HTML object tag to embedded Flash movies. This will restrict the Flash movie and other Flash movies loaded inside it from accessing the dome or from opening sockets.

We also recommend setting minimum cross domain access in the cross domain dot XML file, and trying to avoid wildcard if possible. The last recommendation I can give refers to compiler settings and especially to the Flash player version. It is highly recommended to compile the Flash movie to Flash Player 8 because of the changes in the security scheme that make this version much more secure. As I mentioned, Adobe publishes best practices that should be followed by Flash and Flex developers.

MATHENY: We've covered a lot of information today. Can you summarize Flash and Flex security for us?

YOGEV: Yes. We're explaining the technology and recent research published, Flash and Flex security is becoming a major concern of the security community in the past few years.

Adobe publishes guidelines and best practices according to secure applications, but most Web application developers are not familiar with them, therefore do not use them. There's also a shortage of knowledge and tools. Flash application developers tend not to understand security, and security experts are usually unfamiliar with Flash. This results in many vulnerable Flash applications.

The new version of Apscan developed in Rational introduces complete and comprehensive Flash testing. Apscan dynamically explores all available locations in the Flash movie and tests them for security vulnerabilities. Flash security will become a core part of Apscan, and our security experts are among the leaders of Flash security research done worldwide.

MATHENY: Ayal, this was very informative. Thank you so much for taking time out to discuss Flash and Flex security.

We really appreciate it.

YOGEV: Thank you for listening.

MATHENY: That was Rational's Ayal Yogev, Senior Security Researcher with IBM Rational. If you're interested in more podcasts like this one, check out the Rational Talks To You podcast page at www.ibm.com/rational/podcast. This has been an IBM Rational podcast, I'm Angelique Matheny. Thanks for listening. Keep tuning in as Rational talks to you.

IBM Podcast

[MUSIC] [END OF SEGMENT]