

SEPTEMBER 2003

## Macromedia Product Activation

Macromedia is including a new product activation process in Contribute 2 and its MX 2004 products: Studio, Dreamweaver, Flash, Fireworks, and FreeHand. To support this effort, Macromedia has published information regarding the goals and the implementation of its activation process. @stake conducted research to confirm Macromedia's claims that its products maintain user security and privacy and adhere to industry best practices for product activation technology.

### Overview

Product Activation is a technique used to verify a software product's license prior to the first use of the product. This technique is currently employed by numerous software manufacturers to ensure compliance with Licensing Agreements, but it is new to the Macromedia product line.

Macromedia is aware of the interests and concerns that its customers might have about this feature, so extensive information has been provided to describe the motivation, usage, and technical details of Product Activation. Macromedia contracted @stake to provide an independent, third party verification of the claims made about protecting customer privacy and anonymity.

To verify these claims, @stake was provided access to the release versions of the software. @stake downloaded and installed the software in the same manner as other end user customers. Both Mac OS X and Windows versions were evaluated in a number of different configurations. During installation and activation, @stake closely monitored the software to verify that its behavior was consistent with the documentation provided by Macromedia.

As documented in this report, @stake considered a number of privacy and security concerns raised by Macromedia. In every instance, @stake found that Macromedia's claims were accurate and consistent with industry best practices.

**Claim #1: Internet activation does not collect or transmit any personally identifiable information.**

@stake verified this claim by identifying the source of all information exchanged between each Macromedia application and the Macromedia Activation server.

Fortunately, @stake's testing revealed that Product Activation is implemented very cleanly; the entire set of communications required for Product Activation consists of a single request from a client to the Macromedia server and a single response from the server. This narrowed the investigation considerably.

*Exhibit A* includes a complete Product Activation request captured and decrypted using WebProxy 2 [1].

---

**Exhibit A: Product Activation Request**

--->>--- \*\* https\_3 \*\* Client request to activate.macromedia.com:443 (secure) (04:00:25/11-Jul-2003) --->>---

```
POST /_macromedialicensing/cgi-bin/go.cgi/webstore/XMLActivation HTTP/1.0
User-Agent: MMxpt
Host: activate.macromedia.com
Content-Length: 533
Cache-Control: no-cache
Connection: close
```

```
XML=<?xml version='1.0' encoding='UTF-8' ?>
<LicenseRequest Version='2.0' ClientVersion='3.0.0.105,241000,9.0' LicenseType='SafeCast'
Locking='NodeLock'>
<LicenseFulfillment Type='New'>
<ProductLicenseID>2986577920</ProductLicenseID>
<ProductBuildID>Contribute, en, 2.0, Win</ProductBuildID>
<ReportingDetail>Windows 2000, ENU, 5.0.2195</ReportingDetail>
<ActivationCode>CTD200-79355-97216-90420</ActivationCode>
<ClientData>7322550213856595219531635531509565870998198650962954287203290906</ClientData>
</LicenseFulfillment>
</LicenseRequest>
```

---

It is clear from the captured activation request that the majority of the content does not pose any risk to privacy or security:

- The POST header is of standard format with no client specific information.
- The XML data for the ClientVersion, ProductLicenseID, and ProductBuildID fields include only information about the software being installed.
- As noted by Macromedia in their description, the OS name, version and language are included in ReportingDetail.
- The Activation Code is the serial number provided when the product is purchased.

The ClientData field presents a more difficult challenge, since its contents are not immediately obvious. In order to clarify the content, @stake reverse engineered the software provided by Macromedia to determine the origin of ClientData.

The code used to generate the ClientData is intentionally obfuscated to hinder examination, but @stake isolated many of the key routines. Among the potential inputs used to generate the ClientData are the processor tick-count, system time, local time, process ID, and thread ID. There were variations among the tested applications, however, and @stake was not able to identify every input into the ClientID. When generating the ClientData repeatedly on the same machine, @stake did not observe any structure or repetition in the unknown input values.

The collection of inputs is combined using cryptographic algorithms and the final value is represented in decimal form. Throughout its testing, @stake did not observe any introduction of unique or personal information into the ClientData.

@stake's tests support Macromedia's claim that there is no sensitive or private information within the Product Activation request.

Although privacy concerns are less of an issue for data flowing from the Activation Server to a Macromedia product, it is still worthwhile to examine the server response to identify any concerns.

---

#### Exhibit B: Product Activation Response

---<<<--- \*\* https\_3 \*\* Server response from activate.macromedia.com:443 (secure) ---<<<---

```
HTTP/1.1 200 OK
Date: Fri, 11 Jul 2003 10:58:03 GMT
Server: Apache/1.3.23 (Unix) mod_perl/1.26 mod_ssl/2.8.7 OpenSSL/0.9.6c
Set-Cookie:
MACROMEDIALICENSING_PREFS=1.2%3ACOOKIE_VER52616e646f6d4956f52730b1efe42e4d75db346304
1d586b18ab36c80f2f0f8203e5092671a65bc93ea2eac554daee571946669c917a8de671a2d3814283d9c;
domain=.releasesoftware.com; path=/; expires=Wed, 09 Jul 2008 10:58:17 GMT
Set-Cookie: RELEASE_SESSION=192.168.0.10.94751057921083579; path=/
Content-length: 695
Connection: close
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<LicenseRequest Version="2.0" ClientVersion="3.0.0.105,241000,9.0" LicenseType="SafeCast"
Locking="NodeLock">
<LicenseFulfillment Type="New">
<ProductLicenseID>2986577920</ProductLicenseID>
<ProductBuildID>Contribute, en, 2.0, Win</ProductBuildID>
<ReportingDetail>Windows 2000, ENU, 5.0.2195</ReportingDetail>
<ActivationCode>CTD200-79355-97216-90420</ActivationCode>
<ClientData>7322550213856595219531635531509565870998198650962954287203290906</ClientData>
<License>2152 8485 9710 0355 0403 6067 2522 0111 0181 5323 1032 5129 1591 2255 7921 2125 7286 4571
4013 8035 3211 1227 0746 3794 0518 0693</License>
</LicenseFulfillment>
</LicenseRequest>
<!-- Complete -->
```

---

*Exhibit B* includes the complete Product Activation Response generated by the Activation server and logged by WebProxy following the request in *Exhibit A*. The content within the response is identical to the request with one major exception. The XML content now includes a License value—as should be expected.

**Claim #2: All information transfer regarding Product Activation is protected using SSL.**

@stake monitored all network communication initiated by the Macromedia software during the installation and activation of the product. This showed that all data transfer during product activation was encrypted using SSLv2-TLS. As mentioned by Macromedia in its documentation, this is the preferred encryption for secure web-based transactions.

On Windows, Macromedia's Product Activation uses the wininet.dll libraries for data transfer. This library is distributed with the operating system, and is used by numerous other programs, such as Internet Explorer. This provides assurance that the implementation is correct and that any unique configuration settings, such as a proxy or firewall, are transparently supported.

As Macromedia states in its documentation, the activation request and response are transmitted using e-commerce industry-standard encryption.

**Claim #3: License Transfer does not collect or transmit any personally identifiable information.**

The License Transfer mechanism is another part of Product Activation that requires communication between Macromedia Products and the Macromedia Activation Server. As such, it could present a threat to privacy or anonymity.

Again the simplicity of communication required to transfer licenses makes it possible to verify what information is exchanged. When @stake performed a license transfer, the only communication between each Macromedia product and the Macromedia Activation Server was the request in *Exhibit C*.

---

**Exhibit C: License Transfer Request**

--->>--- \*\* https\_5 \*\* Client request to activate.macromedia.com:443 (secure) (05:03:42/11-Jul-2003) --->>---

```
POST /_macromedialicensing/cgi-bin/go.cgi/webstore/XMLActivation HTTP/1.0
User-Agent: MMxpt
Host: activate.macromedia.com
Content-Length: 538
Cache-Control: no-cache
Connection: close
```

```
XML=<?xml version='1.0' encoding='UTF-8' ?>
<LicenseRequest Version='2.0' ClientVersion='3.0.0.105,241000,9.0' LicenseType='SafeCast'
Locking='NodeLock'>
<LicenseFulfillment Type='Giveback'>
<ProductLicenseID>2986577920</ProductLicenseID>
<ProductBuildID>Contribute, en, 2.0, Win</ProductBuildID>
<ReportingDetail>Windows 2000, ENU, 5.0.2195</ReportingDetail>
<ActivationCode>CTD200-79355-97216-90420</ActivationCode>
<ClientData>2093748477214062099566593896266237971887305918513664114680315762</ClientData>
</LicenseFulfillment>
</LicenseRequest>
```

---

The only difference between this request and the Product Activation Response is that the LicenseFullfillment Type is “Giveback” for a License Transfer Request.

As Macromedia states in its documentation, there is no sensitive or private information within the License Transfer request.

**Claim 4: Telephone Activation does not collect or transmit any personally identifiable information.**

Macromedia applications also allow users to complete Product Activation using the telephone. The Product Activation via telephone uses a subset of the information transmitted during Internet Activation. The data in *Exhibit D* is used to make a phone-based Product Activation request.

---

**Exhibit D: Sample Telephone Product Activation Request**

02 19 03 200 7 9355 97216 90420	-- ActivationCode
21659 38036 12799 42445 35604 98157 51	-- 32 digits of client data
10000140 02	-- 10 digits of version data

---

Although the ActivationCode has a slightly different form, characters A-Z have been replaced with the numbers 0-25 so that they can be entered easily on a numeric phone pad, it still does not reveal any sensitive data. Similarly, version information doesn't pose a risk to privacy.

In order to verify the content within the other 32 digits of the request, @stake reverse engineered the binaries provided by Macromedia to identify the source of the client data. The sources used to generate the data are the same as for the Internet based activation. Because tick count, system time, local time, process ID, and thread ID are not unique, the client data does not present a threat to privacy or anonymity.

**Claim 5: The License Manager only executes while Macromedia applications are executing, uses minimal resources, and uninstalls completely when Macromedia applications are uninstalled.**

Runtime license verification is implemented for both Macintosh and Windows platforms as a service that runs simultaneously alongside Macromedia's applications. @stake monitored the behavior of the license manager throughout all testing. The behavior observed on both operating systems was similar, but the form differs slightly on each.

Both platforms use a collection of data files and registry keys that include binary content representing the license. Although these long-term data stores are accessed repeatedly during execution, their use and structure do not suggest that they contain executable content.

On Mac OS X an SUID binary named “Authentication Service” is executed repeatedly while Contribute is starting. Once a Macromedia application is finished

initializing, the Authentication Service continues to run throughout the life of the application's process, terminating shortly after the application exits.

On the Windows 2000/XP operating system the license manager is installed as a manually started, LocalSystem service named "Macromedia Licensing Service". An additional process related to the license manager is called "~e5d141.tmp". Two instances of this process are present when a Macromedia application is running.

During execution there are also a small number of temporary files created that contain libraries of code used by the Macromedia application and the License Manager. These temporary files were created with a dynamic name when the application began execution, and were deleted by the "~e5d141.tmp" process upon termination.

@stake verified that stopping all Macromedia applications caused the license manager to terminate and temporary library files to be deleted. Further, uninstalling all Macromedia applications resulted in removal of the license manager from the system.

So, @stake's testing matches Macromedia's description of the license manager.

### **Conclusion**

Based on @stake's testing and examination of the Contribute 2 and MX 2004 product binaries, it appears that the claims Macromedia has made about the Product Activation feature are accurate. Further, Macromedia has appropriately minimized the information transmitted during Product Activation so that there is no unnecessary risk to privacy or anonymity.

**About @stake, Inc.**

@stake, Inc., the premier digital consulting firm, provides security services and award-winning products to assess and manage risk in complex enterprise environments. The company's SmartRisk services cover key aspects of security, including applications, critical infrastructure, wireless and wired networks, storage systems, education, and incident readiness. @stake consultants combine technical expertise with a business focus to create comprehensive security solutions for industry leading companies in financial services, information technology, energy & utilities, healthcare, and telecommunications. As the first company to develop an empirical model that measures Return On Security Investment (ROSI), @stake keeps security investments in line with business requirements. Headquartered in Cambridge, MA, @stake has offices in London, New York, Raleigh, San Francisco, and Seattle. For more information, go to [www.atstake.com](http://www.atstake.com).

**Notes and references**

[1] WebProxy is an interactive security tool that helps software developers, quality engineers, and security professionals test and enhance the security of Web applications. See <http://www.atstake.com/webproxy> for more information.

To collect and decrypt the network communications, @stake configured Internet Explorer (and, consequently, wininet) to use WebProxy for both HTTP and HTTPS. Because WebProxy does not have proper credentials for the Macromedia activation site, this initially caused activation to fail. @stake then configured wininet to use the WebProxy self-signed certificate as a trusted authority. Once configured in this manner, product activation was decrypted by WebProxy, and then re-encrypted for transmission to the activation server.

@stake frequently uses this type of testing with secure web applications and it does not represent any threat to users. Because access to the decrypted data requires the client to add a trusted certificate for WebProxy, the decrypted XML content wouldn't be visible in a normal environment.