Realtime publishers

Preventing Data Loss through Unsecure Browsers

The Essentials Series

Preventing Data Leaks from Web Applications

sponsored by

QUARRI™

Dan Sullivan

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

Preventing Data Leaks from Web Applications

Web applications are complex, distributed systems made up of multiple components, each of which requires particular security controls. Servers, network communications, and client devices can all become points of data loss. This article examines the risk of data leaks from Web browsers as well as methods for mitigating those risks.

Threats to Confidentiality and Privacy in Browsers

The browser is a virtually ubiquitous application that makes it an ideal platform for delivering client-side applications. It also makes it an ideal target for someone seeking to steal private and confidential data. Three threats you should consider when designing Web applications are:

- Limits of SSL encryption
- Malware on the client device
- Lures to malicious Web sites

Any one of these could be the foundation for a data leak.

Limits of SSL Encryption

The benefits of encrypting communications are well known: it makes it much more difficult to eavesdrop on a communication session, it provides the ability to authenticate both servers and clients before establishing communication, and from a user's perspective, it is no more difficult to set up a secure channel than an unsecure channel of communication. A significant limit of SSL encryption is that once data is decrypted at a client device, that data becomes vulnerable to attack.

Modern browsers support the SSL protocol (technically, it is the Transport Layer Security Protocol—TLS—but by convention, we refer to it as the SSL or SSL/TLS protocol).

With a secure channel between a server and a client, it is extremely difficult to decrypt a message without the appropriate key. For example, a simple message such as:

> The patent application is attached. Do not circulate outside the company.

Encrypted with one particular key becomes:

```
ftqTSROb9Eua8Rpn1ZA/xCPIl+WfJcUMenvCmFE8bKaRGjASI7KixqAmkk+bFP00w1
o0yBr5lNUQbI2pUhbjgJ4D6mHtcxSSoD2oM8dHvxQ=
```

This is just what we want in a secure communication channel: Even if attackers are able to intercept the messages, they will have no idea what it means. Of course, an application user will have no idea what it means either, so the message must be decrypted before it is displayed to a user or passed to a client-side application for further processing. Unfortunately, once a message is decrypted on the client, it is vulnerable to tampering and copying by malicious software.

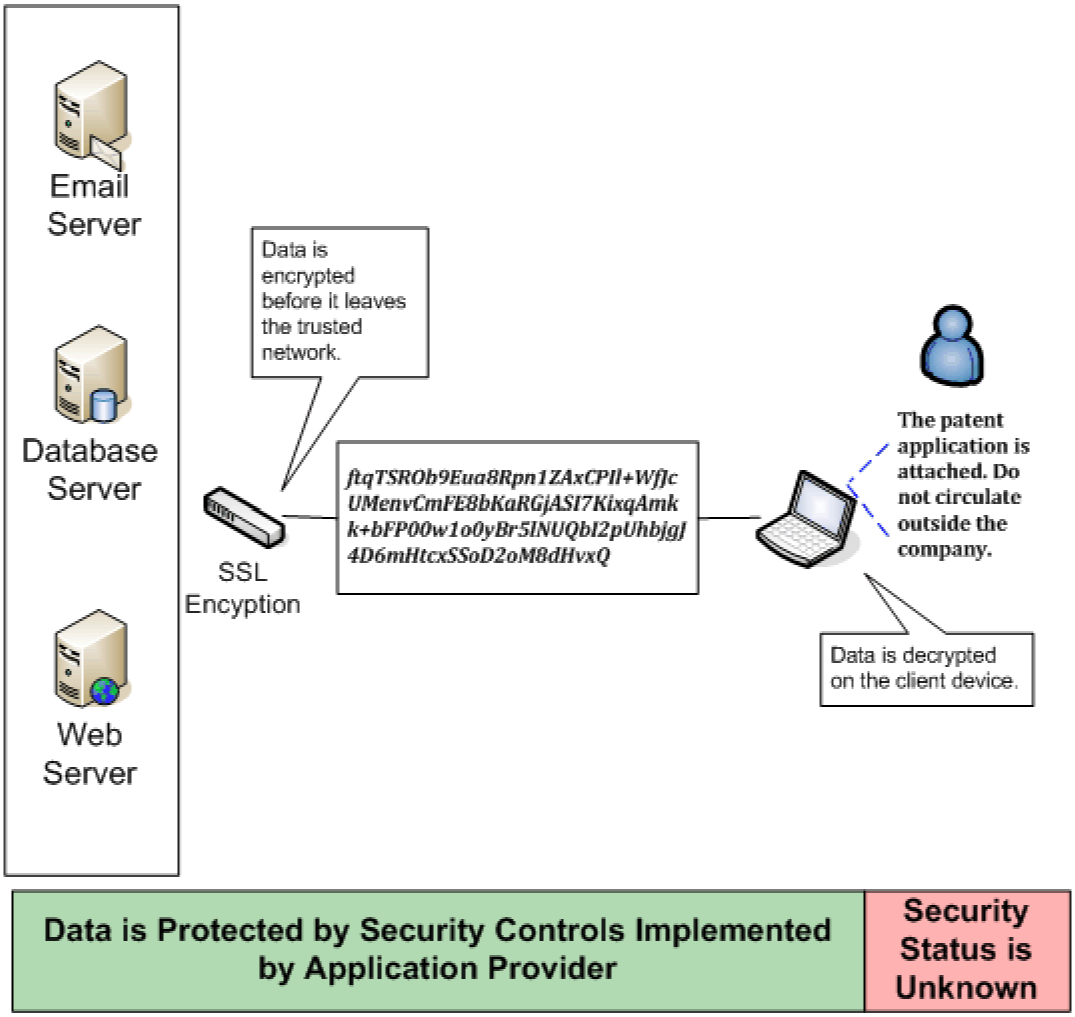
Email Server
Database Server
Web Server
Data is encrypted before it leaves the trusted network.
SSL Encyption
ftqTSROb9Eua8Rpn1ZAxCPIl+WfJc
UMenvCmFE8bKaRGjASI7KixqAmk
k+bFP00w1o0yBr5lNUQbI2pUhbjgJ
4D6mHtcxSSoD2oM8dHvxQ
The patent application is attached. Do not circulate outside the company.
Data is decrypted on the client device.
Data is Protected by Security Controls Implemented by Application Provider
Security Status is Unknown

Figure 1: Even in cases where all traffic is encrypted before it leaves a trusted network, the data must eventually be decrypted on the client device, leaving it vulnerable to tampering or copying.

The security status of a client device may not be known, especially if the device is not managed by the application provider. Consider an employee who logs into the company VPN from a home computer. The employee shares this computer with her teenage children who are adept at downloading everything from music to browser plugins, and, unintentionally, malware. The computer is infected with a malicious program designed to scan storage devices for common office document formats and copy the files to a server controlled by the attacker. The employee may assume it is safe to use the shared computer because her company has well-established security controls, including a VPN. What the employee does not realize is that protection provided by encryption ends as soon as her information is decrypted. This leads us to turn our attention to the kinds of malicious software that can exploit decrypted data.

Malware for Stealing Data

Malicious software may run on either a client device or a server, but both can be designed to steal data that is available in decrypted form within a browser. Some examples include:

- Keyloggers—Client-resident malware that capture keystrokes as they are typed
- Video frame grabbers—Client-resident malware that copies data from memory used to display information on a monitor
- Malicious browser add-ons—Malware that takes a Trojan Horse approach and appears to do one thing, such as provide weather reports in the footer of the browser, while stealing data from the browser cache
- Hostile SSL proxies—Services that appear like legitimate encryption and proxy services but implement a man-in-the-middle-attack allowing a third party to intercept and decrypt your communications
- Injection attacks into browser process spaces—Malware that takes advantage of vulnerabilities in application code to force the client application to return data in ways that was not intended by the application designer

Imagine using an online banking service from a compromised device. A keylogger could capture information you type, such as account numbers; a video frame grabber could copy full screen images, including lists of transactions; and a malicious browser add-on might copy a bank statement you download in the form of a spreadsheet. All of these forms of attack are available for stealing data because decrypted data on a client device is not sufficiently protected by Web browsers.

Lures to Malicious Web Sites

Application developers and managers appreciate the benefit of deploying applications through a browser, including the fact that you do not need to install software on client devices. Malware developers can take the same approach: Why bother tricking a user into downloading malware that might be detected by antivirus software when you can lure them to a Web site hosting malicious software? Phishing scams, fake messages from IT security departments, and promises of free or heavily discounted products and other lures can draw unsuspecting users to sites designed to steal information.

The limits of SSL encryption, the potential for malware on client devices, and the threat of lures to malicious Web sites can all compromise the security of a Web application. Clearly, more must be done to secure information in Web browsers.

Securing Information in the Browser

There are several requirements to protect sensitive data on the client devices:

- Ensuring that decrypted data is only be available to client-side application code related to the Web application that delivered the data
- Logically isolating data sent to the browser by the Web application from other processes
- Deleting cached and downloaded data when a session terminates
- Controlling and auditing user activities in the browser (for example, cut and paste) to prevent malicious activities by the user, such as copying customer lists or trade secrets

To meet these requirements, you need to build Web applications using security software that can be delivered to the client device on demand.

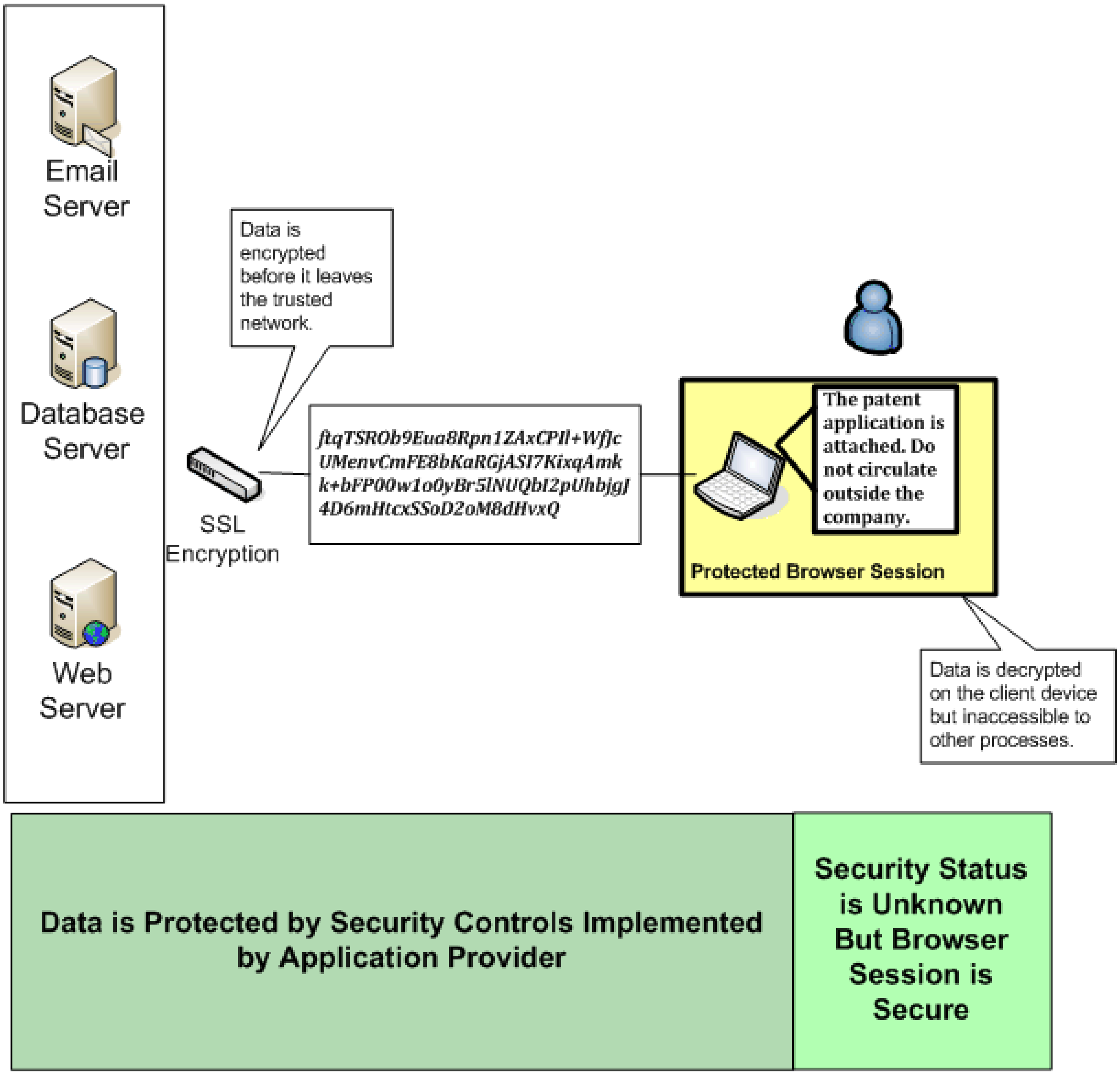
Email Server
Database Server
Web Server
Data is encrypted before it leaves the trusted network.
SSL Encryption
ftqTSROb9Eua8Rpn1ZAxCPIl+WfJc
UMenvCmFE8bKaRGjASI7KixqAmk
k+bFP00w1o0yBr5lNUQbI2pUhbjgJ
4D6mHtcxSSoD2oM8dHvxQ
The patent application is attached. Do not circulate outside the company.
Protected Browser Session
Data is decrypted on the client device but inaccessible to other processes.
Data is Protected by Security Controls Implemented by Application Provider
Security Status is Unknown But Browser Session is Secure

Figure 2: The risks to decrypted data on the client can be mitigated by creating a secure browser session that protects application information from other potentially malicious applications on the client device.

You cannot assume any client device will meet an application's security requirements. The security software will have to essentially build a logical barrier around all session data and monitor client-side activities, such as copying and pasting. On-demand client-side security controls are available today, making available an additional level of protection to application developers.

Summary

Regardless of the security measures taken to protect data on servers and during transmission to client devices, Web applications are potential points of data loss because of weaknesses in Web browsers. SSL encryption helps protect data when it is sent between clients and servers, but once it is on the client device, the data must be decrypted to be useful to the end user. This situation creates an opportunity for malicious software on the client device or downloaded from a compromised Web site to steal decrypted data.

Mitigating the risk of data loss from a Web browser requires additional security measures to the ones you typically deploy. Web applications that work with confidential and private information should not assume that client devices are secure. Instead, they should download browser session security software prior to launching the Web application's client-side code. Session securing software should provide a logical barrier from other processes on the client so that malicious software cannot read from the browser's applications space. In addition, the session securing software should delete all session data after the Web application terminates so that no data is left in the browser cache, which could be vulnerable to malware on the client. Of course, not all data loss risks stem from malicious software. A disgruntled employee who decides to download a client list and take it to a new employer can be more of a threat than malware. The session securing software should include functionality that controls user activities and logs user actions. This information may be useful for enforcement and compliance.

IT professionals have put a great deal of effort into securing Web applications. Until recently, you have had to count on users to employ caution when browsing and clearing sensitive data from the browser. You've also had to assume and hope that users kept their antivirus and personal firewalls up to date to mitigate the risk of a malware infection. Fortunately, you do not have to simply hope for the best anymore. On-demand, session-based security controls now offer the opportunity to extend security controls to even unmanaged client devices.