

Realtime
publishers

"Leading the Conversation"

The Essentials Series

Messaging and Web Security Volume III

by Dan Sullivan

Article 25: Reducing the Risk of Insider Abuse	1
Classifying Data.....	2
Principle of Least Privilege.....	3
Principle of Separation of Duties.....	3
Rotation of Duties	4
Summary	4
Article 26: Anti-Forensics: Digital Investigations Get More Difficult.....	5
Basics of Computer Forensics	5
Anti-Forensic Techniques.....	6
Summary.....	6
Article 27: Systems Recovery, Virtual Images, and Keeping Security Measures Updated	7
Limitations of Common Virtual Image Practices	7
Options for Maintaining Virtual Images.....	8
Article 28: Security and the Software Development Life Cycle	9
Requirements Gathering	9
Design	10
Development.....	10
Testing.....	11
Summary	11
Article 29: Browser Sniffing Attacks and What to Do About Them.....	12
Motivation for Browser Sniffing Attacks	12
Browser Cache Sniffing and Related Techniques	13
Reducing Risk of Browser Cache Sniffing.....	13
Summary	14
Article 30: Green Computing and Security	15
What Is Green Computing?	15
How Virtualization Can Improve Security	16
Power Management and Security Benefits.....	16
Proper e-Waste Disposal.....	17
Summary	17

Copyright Statement

© 2008 Realtime Publishers, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers, Inc or its web site sponsors. In no event shall Realtime Publishers, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

Article 25: Reducing the Risk of Insider Abuse

In the summer of 2008, the city of San Francisco found itself the victim of an apparent case of insider abuse. A city employee allegedly locked out other administrators from the city's fiber optic network and reserved-top level administrative privileges for himself. The employee was arrested and eventually surrendered passwords to the mayor of San Francisco. Many might wonder how an organization the size of the city of San Francisco, with complex networks and information systems, could become victim to a single person? Unfortunately, it is easier than we might imagine.

Insiders, including employees, contractors, and consultants, have access to information and systems that can be used to the benefit or detriment of a company. Obviously, we cannot restrict IT professionals too severely or they cannot accomplish their tasks; at the same time, too much trust in any one individual can place an organization at inordinate risk. The solution, as is often the case in information security, is to find a proper balance between competing needs. It also requires structuring responsibilities in such a way that reduces the chances that a single individual could cause a serious disruption to the organization. Four common components of a balanced approach to reducing the risk of insider abuse are

- Classifying data
- Employing the principle of least privilege
- Employing the principle of separation of duties
- Rotating duties

The first component helps an organization to understand the most valuable assets that demand protection; the others are management practices designed to help reduce the risk to those assets.

Classifying Data

Organizations generate and manage increasing amounts of heterogeneous data. Just about any business will have financial, human resources, customer management, operational, and project management data. This data often resides on different servers running different applications. Sometimes the data is duplicated and shared across devices; some may even land on unmanaged devices such as employees' laptops or smartphones. The first step to managing risk to data is to classify it so that resources are appropriately allocated to protect it.

One classification scheme includes four categories:

- **Public data**—This data is available to all, and includes information such as press releases and publically available government filings. It may also include documents and data that might not be intended for public review but would not adversely affect the organization in any way if it were.
- **Sensitive data**—This type of data should be shared only within the organization, but if it were released, it would not cause severe harm to the organization. Information such as project plans, email exchanges between employees on operational issues, and status memos might fall into this category.
- **Private data**—This type of data about employees or customers should not be released to the public. Examples include employee performance reviews, customer credit card information, and payroll information.
- **Confidential data**—This data should be limited to employees and business partners under non-disclosure agreements. Any information that would provide competitors with trade secrets, intellectual property, or other information that would jeopardize a competitive advantage or disclose strategic plans falls into this category.

With regards to insider abuse, it is important to limit access to private and confidential information to those employees or business partners that need such access for their jobs. For example, a technology consultant may be under a non-disclosure agreement, but unless she is working on a human resources project, she should not have access to payroll information. Similarly, an employee in the finance department should not have access to meeting notes from executive strategy sessions.

When data is properly classified, you can determine appropriate access levels for systems as well. Jobs and roles in an organization can be mapped to particular types of data access requirements, which, in turn, dictate which applications and servers a person needs to access.

Principle of Least Privilege

Limiting access to data and applications is one way of enforcing the principle of least privilege. This principle states that a person should have all the access to data and applications needed to do a job but no more. For example, a finance department employee who manages accounts receivables should not have access to all functions in a financial management system. Similarly, a service desk technician should have the ability to reset passwords and install software on other employees' workstations but should not have access to a database server that is managed by a database administrator (DBA).

Role-based access controls help enforce the principle of least privilege. One can craft fine-grained access rules according to the particular needs of an organization. For example, within a database, users can be granted read, write, update, and delete privileges to some tables, read access to others, and no access to still others. This principle helps reduce the likelihood of insider abuse by containing the damage that a single person can inflict.

Principle of Separation of Duties

The idea of separation of duties is that no critical function should be accomplished by a single person. Some businesses, for example, require two authorizing signatures on checks over a certain amount to reduce the risk of embezzlement. In IT, separation of duties is important to prevent the kind incident that the city of San Francisco suffered. Other areas of information technology (IT) management demand separation of duties as well:

- Developers and testers should be different individuals to ensure programs are properly tested before being released into production.
- DBAs should not have full administrator privileges to database servers to ensure they cannot tamper with log files that might indicate suspicious changes to a database.
- Individuals that authorize changes to routers or other network devices should not be the same person that implements those changes; otherwise, changes can be made secretly.

Separation of duties reduces the risk of insider abuse by limiting one person's ability to execute the full sequence of events required to damage a system without detection.

Rotation of Duties

It is a sound practice to have multiple employees cross trained in different tasks. If a key person is on vacation, suddenly becomes sick, or leaves the organization, it pays to have someone else ready to step in immediately. Regularly rotating duties also helps reduce that chance that someone can hide unauthorized activities. For example, if someone in the finance department found a way to extract customer credit card information and sell it on the underground market, that person's activities might be detected when another person is assigned their responsibilities and given their privileges. They newly assigned employee may find indications of unauthorized jobs running in the middle of the night, suspicious file transfers, or inordinate disk space use.

The object of this practice is to have a number of technically knowledgeable individuals in charge of vulnerable processes. In this way, no one person can alter the operation to their personal advantage or to the detriment of the organization.

Summary

The potential for insider abuse cannot be eliminated completely, but the steps outlined in this article can reduce the potential for such abuse. For more information about reducing this risk and for examples of incompatible functions across a wide range of IT functions, see CISA [Review Manual, 2008 Segregation of Duties Control Matrix](#).

Article 26: Anti-Forensics: Digital Investigations Get More Difficult

Anti-forensics is the practice of hiding evidence of a security breach to prevent detection of the breach and its perpetrators. It is now a major problem for computer security investigators.

When a security breach occurs, a common first reaction is to contain the damage by isolating the compromised device and blocking network access. The next major step is to get the system restored to its pre-attack state and functioning normally. Then investigators can turn their attention to analyzing how the breach occurred, but will it do any good? Computer forensics, the practice of analyzing digital information to determine the way a computer crime was committed, is facing serious challenges from attackers using anti-forensic techniques. This article briefly reviews forensic techniques and describes how readily available tools are rendering these techniques far less effective than they have been in the past.

Basics of Computer Forensics

The goal of forensics is to discover how a security breach occurred and what data, configurations, programs, or other elements of a system were changed during the breach. Data may be collected in real time if an attack is detected while in progress, or after the fact using evidence left in memory or on disk drives.

Computer forensics begins with a data collection process. Assuming an attack has occurred and is no longer in progress, an investigator will make an electronic copy of data on compromised systems. Since digital evidence is easily changed, it is better to analyze copies rather than risk accidentally changing the original. Also, data may be collected from multiple sources, such as disk drives, memory, and removable storage devices. User files, operating system (OS) logs, database files, and other types of data may be collected. Investigators may calculate hash values for each piece of evidence to ensure any change to the data after the attack would be easily detected.

After collection is complete, analysis begins. This stage of the investigation employs a wide range of techniques looking for information about activity on a system. This includes analyzing

- File metadata
- Data blocks on the file system
- File system storage information
- Hidden protected areas on disk drives

The objective of this phase is to detect digital traces of activity on a system. For example, when a file is modified, the OS will update the modified date in the file header. If an attacker installs malware, under normal circumstances, there will be a new file created to store the file. The problem for forensic investigators today is that relatively easy-to-use tools are readily available to hide or modify the data investigators depend on.

Anti-Forensic Techniques

Anti-forensics is the practice of tampering with or hiding digital evidence of a security breach. Some ways this is done includes:

- Overwrite data
- Modify timestamps
- Modify file header information and extensions
- Hide data in allocated but unused data blocks at the end of files
- Randomly generate file names for malicious code to avoid detection

One of the simplest ways to remove evidence is to overwrite storage devices or files. This requires more than simply deleting a file, which typically changes an entry in a disk management data to indicate the space occupied by the file is available for reuse. One could delete a file but the contents of that file can remain on the disk until the storage is overwritten. (This feature of disk storage is exploited by “undelete” utilities.) Overwriting data blocks with random data several times virtually ensures that the data once there is irrecoverable.

Forensic investigators can piece together parts of the sequence of events in an attack if accurate timestamps are available. Although the OS will update timestamps when a file is changed using OS functions, the data can also be updated by anti-forensic tools, leaving investigators with useless information.

In addition to timestamps, other information in the file header may be tampered with to throw off investigators. For example, an executable may be masked as an image file so that it is not easily detected.

Another anti-forensic technique takes advantage of the way file systems allocate storage. To make storage and block handling more efficient, standard block sizes are used. As files grow, the last data block is written to; once it is full, a new block is added. The last block in a file may have unused space that is used by attackers to hide data. This method helps obfuscate the attackers’ activities by not allocating any new storage from the file system.


Signature-based detection is a well-established method for detecting viruses and other malware. To avoid this kind of detection, attackers may use random filename generators. Changing filenames still leaves files susceptible to detection based on hash values, but this can be avoided as well by adding random data to a file in such a way that does not disrupt how the file is used.

Summary

Forensic techniques have been used to uncover information about attacks, but the advent of easy-to-use anti-forensic tools is reducing the utility of forensic techniques. As detecting and prosecuting attackers becomes more difficult, there is even more reason to pursue defense-in-depth measures that block attacks.

Article 27: Systems Recovery, Virtual Images, and Keeping Security Measures Updated

Server virtualization has brought greater efficiency to data centers, but it demands changes to the way you manage infrastructure. In terms of benefits, server virtualization has reduced the number of physical servers you need to rack mount in your server rooms, reducing capital costs as well as operating costs. One of the new management tasks is the need to create and maintain virtual images—and this is where new procedures are needed.

 Storage virtualization is another boon for IT efficiency, but that is outside the scope of this article; here, the focus is on server virtualization.

Limitations of Common Virtual Image Practices

Consider a couple of representative scenarios for the need for short-term virtual servers. For example, suppose developers need a Linux host to test a new database application. No problem, create a virtual server. Systems administrators need to do some planned maintenance on an application server? Create a virtual instance of the application server on another physical server; performance may degrade, but business operations can continue.

To maximize the efficiency and consistency of virtual servers, virtual images are created with standard configurations. These can include the operating system (OS), drivers, and applications needed for particular applications. Systems administrators, for example, could create a standard Windows Vista image for desktop clients, a Windows Server image for an email server, and a third image with Linux/Apache/MySQL/Perl (LAMP) systems for developers. When another virtual server is needed, one of these images can be pulled down off the shelf and deployed to a physical server.

However, problems can arise over time as these images become outdated. This can happen for several reasons. For example, the software included in an image may have been upgraded or patched since the image was made. Microsoft, for example, provides monthly updates on “patch Tuesdays,” while Oracle uses a quarterly release schedule for its patches. Other software providers, such as the Mozilla Foundation, provide updates for Firefox on an as-needed basis. Systems that are online can retrieve patches automatically from a vendor or patch server and apply them according to an organization’s policy or leave it to users to manually install updates. These options, however, are not available for keeping virtual images up to date.

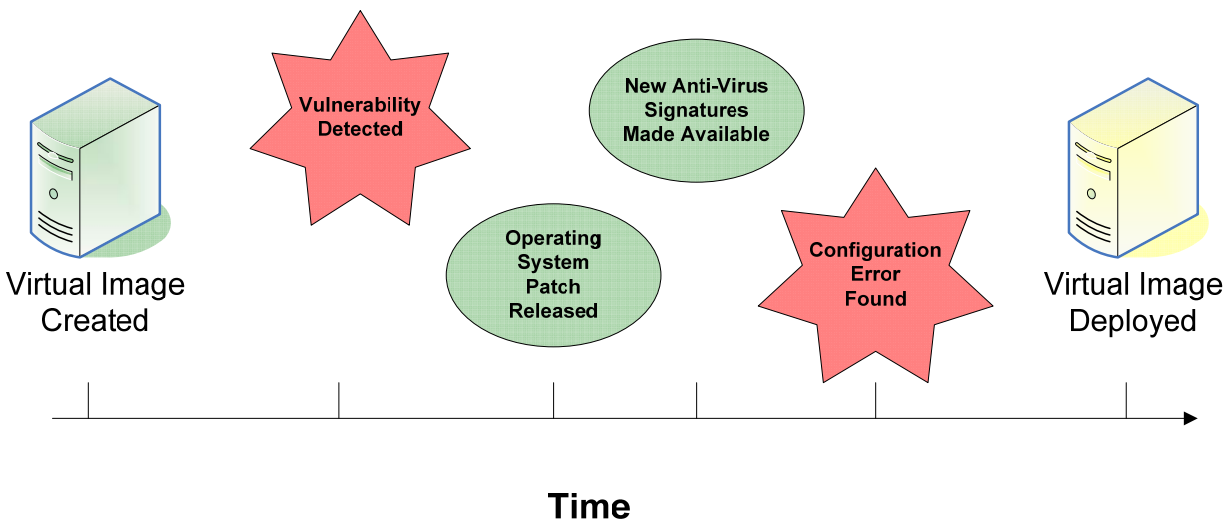


Figure 1: Virtual images can lag behind in patches, updates, and other corrections, resulting in virtual servers that are out of date when they are deployed.

Options for Maintaining Virtual Images

There are a number of ways to avoid deploying out-of-date virtual images. Each of these, or some combination of these, can be the best option in different scenarios.

When images are infrequently used, it may be best to generate the virtual image when it is needed. This option has low maintenance overhead but can require more time to create the image when it is needed. The image is generated on demand, so there is no risk of deploying a stale, out-of-date image.

Another option is to maintain a library of images and scan them using antivirus programs on a regular basis. An image might harbor undetectable malware at the time of creation but an update to the antivirus system may detect and remove it at a later time. When the image is deployed, it will be free of detectable malware and the signature files and antivirus program can be updated immediately.

The advantage here is that standardized virtual images can be created and stored with some assurance that, from a malware-detection perspective, the image will be as updated as online servers. This approach also reduces the time to deploy a virtual server because the image does not need to be generated on demand.

A third approach is to maintain copies of patches, updates, and modified configuration files along with virtual images. When a virtual image is deployed, the patches, updates, and modified configuration files can be applied and installed. When this procedure is applied methodically, we can be sure that images are up to date when they are released into production.

Server virtualization can improve the efficiency of IT operations, but we must adopt operational procedures to ensure we do not undermine the advantages of virtualization by allowing malware, vulnerable software, or misconfigured applications into the IT infrastructure.

Article 28: Security and the Software Development Life Cycle

Software developers, in many ways, are the start of fundamental security strategies. You can apply countermeasures to networks, lock down servers, and monitor and audit everything that moves, but if the software you run is flawed, the demands on these security measures increase. Security is not a feature that can be added late in the software development life cycle anymore than the structural integrity of a building can be added like another room on a house. It is imperative that security be incorporated at the start of the life cycle. This article outlines essential issues to address at major steps of the software development life cycle, including:

- Requirements gathering
- Design
- Development
- Testing

Each stage of the software development life cycle has distinct characteristics and provides unique opportunities for improving the overall security of an application.

Requirements Gathering

Software development begins with understanding business requirements, including requirements related to protecting the integrity, confidentiality, and availability of data. At this stage of the life cycle, one does not try to address how the security issues will be addressed but to simply enumerate them. For example, during requirements gathering, analysts should determine:

- Which application functions will be available to which users?
- What roles are required to organize user groups and allocate privileges accordingly?
- What governance and compliance policies apply to the application and its data?
- What are the consequences of unauthorized use of the application or disclosure of data?

Ideally, at the end of requirements gathering stage, you will have an understanding of the value of the application and data as well as the risks to those assets.

Design

During the design stage, requirements are mapped to an implementation model. This process includes both application and database design. The key issues that are addressed in this stage are:

- Designing data storage and data flows in a secure manner. For example, confidential data should be encrypted during transmission.
- The authentication mechanisms should be identified. This includes users authenticating to the application and the application authenticating to services used by the application. For example, calls to Web services may require the use of WS-Security standards.
- Defining design principles, such as the use of parameters to exchange information rather than global data stores. Modularized code is easier to maintain and debug when shared global data structures are not used.
- Process flows should be documented and analyzed for vulnerabilities. For example, is it possible to bypass an authentication check and query a database directly?
- Application programming interfaces (APIs) should be designed with security in mind. The data access interface, for example, may consist of a set of services or functions that return data rather than allowing application developers to issue SQL queries. This makes it easier to properly code SQL queries to prevent SQL injection attacks and allows the DBA to assign SELECT, UPDATE, INSERT, and DELETE privileges only to these functions and not to users.

The security focus at the design stage is on minimizing architectural vulnerabilities and using defensive strategies, such as data access APIs, to reduce the chances of introducing vulnerabilities during the development phase.

Development

By the time you are in the development phase, you should be done addressing architectural issues; the focus should shift to secure coding practices. In general, this includes:

- Writing robust error handlers to reduce the chances of putting the program in a state that could be exploited
- Using analysis tools to find errors such as a lack of bounds checking
- Always validating input, especially if the input is used directly in a database query
- When in doubt, denying access to data and functions
- Keeping code as simple as possible to accomplish a function; complex code is more difficult to maintain and more likely to introduce vulnerabilities than limited code

Practicing secure coding techniques will help reduce the risk of introducing vulnerabilities, but there are no guarantees, and that is why you test.

Testing

Sound testing plans always include comprehensive functional testing, but it should also include security testing. Testing should involve a wide variety of input sets, including inputs outside the normal range. For example, if an input up to 30 characters is expected, test with a 50-character input string; if a database lookup is expecting a numeric entry, test with an alphabetic input.

Summary

Software designers and developers are the first line of defense in information security. Well-designed, properly coded, and thoroughly tested code can reduce vulnerabilities in systems and make significant contributions to the overall security of an IT infrastructure.

Article 29: Browser Sniffing Attacks and What to Do About Them

A browser cache contains a historical record of where we have recently been and what we have recently done on the Web. Web browsing performance is much better with caching; some estimate that Web caches can service up to 60% of requests (Source: Collin Jackson, Andrew Bortz, Dan Boneh, and John C. Mitchell, “[Protecting Browser State from Web Privacy Attacks](#),” WWW 2006. There are security costs associated with this performance improvement, though. We leave a trail of banks we conduct businesses with, the social networks we use, the collaboration tools we work with, the email systems we use, and a wealth of other personal information that can compromise our privacy.

This article will discuss motivators for attackers to go to the trouble of stealing this information, ways they can get at it, and simple practices that can help reduce the chances attackers will find useful information in your browser cache.

Motivation for Browser Sniffing Attacks

Phishing is getting more difficult. Email users are becoming increasingly aware of spam and phishing attacks. Common scam lines have reached so many of us that requests for funds from political exiles in return for enormous returns is now better fodder for late night comics than actually luring a victim. Email filters are also adept at identifying and filtering spam messages. Although promising, neither of these trends will eliminate phishing; they simply put pressure on phishers to develop new techniques.

One of the key hurdles phishers have to overcome is making a phishing lure—the spam message that gets a victim to link to a malicious site—appear legitimate. Generalized pleas cannot be counted on. Masquerading as a large bank and sending out warnings about the victim’s bank account under the assumption that some of the recipients will actually be customers of the bank are likely to underperform if they manage to make it past the email filters. Customizing an email message with details about the victim is more likely to build credibility and trust than generic content. These types of messages are also less likely to be caught by statistical pattern recognition email filters. The challenge for phishers then becomes how to collect the needed personal data—this is where browser sniffing comes in.

Browser caches are unique repositories of personal information. The data is broad but not detailed. For example, it may contain information about which bank you use and where you shop but not your credit card data. With browser cache data alone, attackers may not be able to commit fraud or steal you identity immediately, but it is a step toward that goal.

Browser Cache Sniffing and Related Techniques

There are at least a few ways to steal information from browser caches, and related information such as bookmarks. One technique demonstrated by researchers is to lure a user to a Web site that then retrieves Web pages from sites of interest, such as banks, brokers, and retailers. As those pages are retrieved, a script is running to time the how long it takes to retrieve; pages with short times relative to other pages are likely to have been in the cache. For example, a malicious site might request the login page for 10 of the top-10 banks in the country. Presumably, if one page is retrieved significantly faster than the others, that is the user's bank. Similarly, a rapid response to a Web retailer's site might indicate a recent purchase. These pieces of information can be used to craft a phishing lure about bank account security or a problem with a payment.

Another technique takes advantage of browser history tracking. Links to visited sites are often displayed in a different color than links to sites that have not been visited. Visited sites have a pseudo-class attribute “:visited” associated with them to allow for this formatting distinction. That same information can be used to determine previously visited sites for malicious purposes.

A third technique exploits cooperative tracking. In this technique, Web sites coordinate their tracking of information about users' visits. Of course, legitimate sites are not going to knowingly coordinate tracking with malicious sites, but attackers can take advantage of cross-site scripting vulnerabilities in Web sites to implement a de facto cooperative tracking.

Reducing Risk of Browser Cache Sniffing

The best way to reduce the risk of browser cache sniffing is to minimize the amount of information in the browser. There are several ways to do this.

First, you can manually clear the cache. This is a simple and effective technique but it requires users to remember to clear the cache and to take the time to navigate through browser menus to execute the command (See Figure 1).

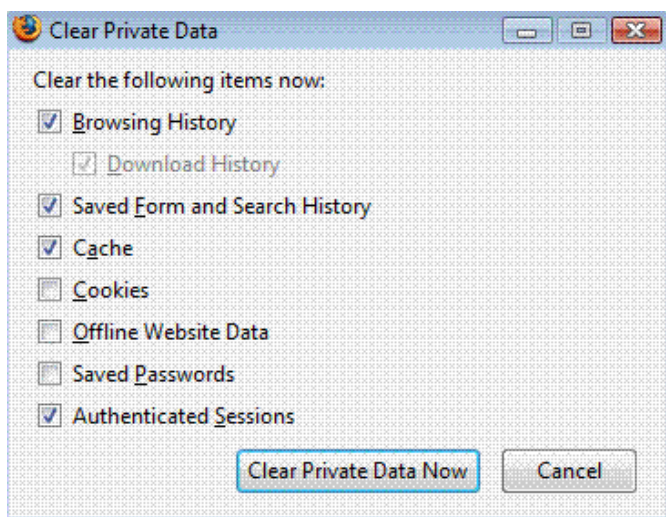


Figure 1: The browser cache can be cleared manually by using browser-specific commands, such as these in Mozilla Firefox.

Alternatively, you can use browser plugins, such as Distrust for Mozilla Firefox (available at <https://addons.mozilla.org/en-US/firefox/addon/1559>). This add-on allows users to indicate at the start of a “distrust session” in which all activities are monitored. When the session is ended, all cookies and cache entries created during the distrust session are removed. Note, that add-ons are not necessarily supported as well as the browser they work in. For example, SafeCache, a similar add-on to Distrust, is supported in Firefox 2 but not Firefox 3. Add-ons should be tested in an appropriate environment before being used in a production environment.

Another way to remove browser information is with a general privacy protection/system cleaning tool. These programs are often bundled with other security programs, such as antivirus and firewalls, and give users the means to quickly purge the browser cache while also removing temporary files and history information and clean up registry entries (see Figure 2).

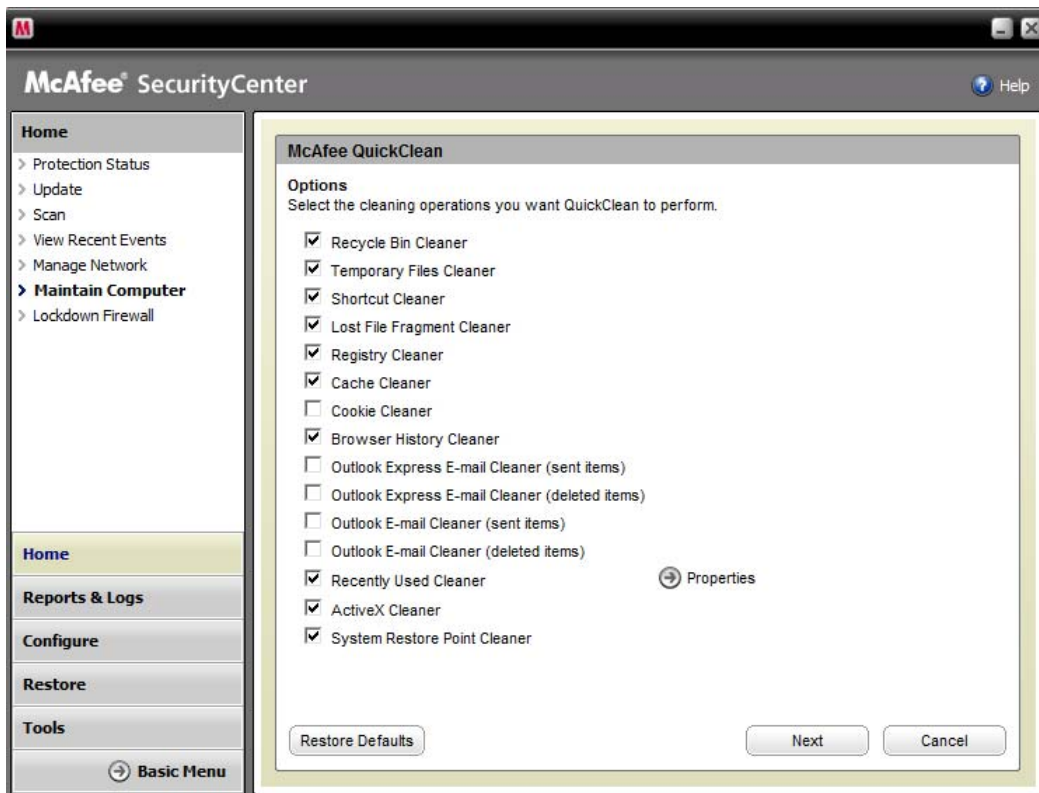


Figure 2: General privacy protection and system cleaning tools can clear the browser cache as well as purge other potentially private information.

Summary

Browser cache sniffing is one way to collect private information about users for targeted attacks. The benefits of using the browser cache typically outweigh the security disadvantages. Nonetheless, additional measures can be taken to reduce the risk of leaking private information.

Article 30: Green Computing and Security

Green computing is a broad collection of practices designed to reduce the environmental impact of computing, but its benefits extend beyond that. Businesses find the efficiencies of green computing translate into cost savings, and in some cases, improved security. This article begins with a brief discussion of green computing practices and then describes ways these practices improve security, from protecting servers to managing the end-of-life stage of computer equipment. Green computing is not just for the environmentally concerned; it is an additional method to cut cost and strengthen security.

What Is Green Computing?

Green computing encompasses a wide range of techniques to lower the impact of computing on the environment over the full life cycle of computing systems. It involves:

- Redesigning computing devices so that they are easier to disassemble and recycle; this also includes using lower-impact chemicals in the manufacture and recycling process.
- Designing devices to reduce power consumption through a combination of re-engineered devices that use less power, software to measure power consumption, and better data center design to reduce cooling costs.
- Consolidating computing services through virtualization. Wasted CPU cycles consume power but with virtualization, companies maximize the utilization of multi-core processors while reducing the number of physical servers needed to provide the required level of service.
- Educating users on energy-saving practices, such as powering down workstations at night and consolidating the number of printers in an office.
- Disposing of electronic waste (e-waste) in a way that minimizes hazardous chemicals released into the environment and reuses materials when feasible.

From this list, virtualization, power management, and proper e-waste disposal can all have a beneficial impact on information security.

How Virtualization Can Improve Security

Operating system (OS) virtualization offers a number of security benefits. Systems administrators can create virtual images and instantiate them multiple times. Doing so reduces the chance that an OS is misconfigured or a patch is missed when installing a system. A single image can be used over and over again, so there is more time and incentive to properly configure the image initially. More time can be taken to research vulnerabilities, ensure only necessary services are running, and install the latest security software.

Replacing an OS in a virtual environment can be as simple as shutting down a virtual instance and restarting with a new image. This process can streamline patching. Administrators can apply patches to virtual machine images, copy the image to a physical server, and restart. Pushing known-good virtual machine images to devices on a frequent schedule can help reduce the impact if a deployed virtual machine instance does become infected with malware. Even if the malware were undetected in the deployed instance, replacing the virtual machine image eliminates the problem.

Another advantage comes from more efficient antivirus scanning. Some antivirus vendors now offer applications that scan virtual machine images. These applications can be used to scan the source copies of virtual machine images each time updates are made to the antivirus software. If malware is detected, it can be removed and a new image deployed.

When virtual machines are deployed into a cloud environment, such as the Amazon Elastic Computing service, there are cost incentives to shut down the virtual machines when they are not needed. A virtual machine that is not running will not be a vulnerable target, and if the virtual machine had been compromised, it would be of no use to attackers.

Power Management and Security Benefits

Power management practices, such as shutting down desktop PCs, can improve security as well. As with virtual devices, when hardware is powered down, it is not susceptible to common attacks. Device management hardware, such as functions included in the Intel vPro line of processors, enable remote shutdown and startup. With this functionality, employees can shut down their workstations at the end of the day, and those workstations can be automatically powered up by asset management software when patches need to be pushed to the device.

If a device has been compromised, for example, if it has been infected with a botnet application, powering down the device will prevent it from generating spam, launching Denial of Service (DoS) attacks, or be used for other malicious purposes.

Proper e-Waste Disposal

There was a time when someone might simply dump an old PC into the dumpster at the end of the computer's useful life. Those days are gone. Security concerns and compliance requirements dictate proper management of end-of-life cycle devices.

Electronic devices, especially hard disk drives, can contain too much confidential and proprietary information to simply throw away the device. File shredders and disk over-writing utilities such as McAfee Quick Clean or Darik's Boot and Nuke (DBAN) can prevent others from recovering data from old drives. Responsible e-waste recyclers may wipe drives to U.S. Department of Defense (DoD) standards or the machines can be wiped before the machines are sent to recycling. Green computing has brought greater attention to end-of-life cycle devices and the need for proper disposal of data and hardware.

Summary

Green computing may have started with environmental concerns but the benefits of following green practices extend to information security. Virtualization, power management, and proper e-waste disposal overlap the concerns of sound environmental practices and good security procedures.