BigFix Patch Manager
Functional Overview

Network administrators face a dramatic surge in the number of security threats and are struggling to maintain secure systems while controlling operational cost. According to the Computer Emergency Response Team (CERT), more than 4,000 vulnerabilities were reported in 2002. In just the first quarter of 2003, more than 900 vulnerabilities were reported, representing an average of 10 new potential problems every day. Yet most attacks exploit known vulnerabilities and can be prevented by proactively applying security patches as soon as they become available.

On January 24th, 2003, the SQL Slammer worm exploited a vulnerability announced just 6 months earlier. On August 11, 2003, the Blaster worm exploited a vulnerability announced just one *month* earlier. In both cases, a patch was released at the time the vulnerability was announced. Slammer and Blaster effectively demonstrated that timely and effective management of security patches is of the essence to maintaining secure systems.

A manual approach to managing security patches and updates includes regular hands-on sweeps of selected security sites for vulnerabilities, assessment to determine if the network is vulnerable, analysis and development of a remediation, and then deployment of the remediation to thousands or tens of thousands of computers throughout the enterprise. The manual approach is labor intensive, costly, error-prone, and slow. Automated patch management is the best antidote to the increasing numbers and pace of security threats in large networks.

Selecting an Effective Automated Patch Management Solution

Network administrators recognize the need for a flexible, easy-to-use system that: proactively detects the vulnerabilities in their networks; provides them with enough information to make educated decisions about remediation; eliminates the vulnerabilities quickly; and, automatically alerts them if the vulnerabilities ever re-appear. Evaluating automated patch management solutions involves not only evaluating these capabilities, but also the architecture and security of the solution. Table 1 summarizes key patch management feature and architectural requirements.

Feature Requirements	Architecture Requirements
• Includes a complete and up-to-date patch library • Quickly and accurately determines what patches are relevant and for which machines • Supports test deployment in lab environments prior to mass deployment • Provides an intuitive administrative interface to sort and prioritize patches • Effectively targets deployment to single devices or groups of devices based on network architecture • Gives complete control over deployment scheduling, reboots and user messaging • Supports policy creation to ensure mandatory patches are deployed and stay deployed • Provides tools to customize and create custom patches • Provides comprehensive, real-time reporting	• Works effectively in large networks without degrading network performance • Supports 15,000 clients per server to reduce incremental hardware investment • Supports heterogeneous environments • Optimizes deployment over low-bandwidth connections with download restart and bandwidth throttling • Utilizes a “relay” topology to maximize speed of administration for highly-distributed networks • Integrates with Active Directory and other directory services databases • Protects system from unauthorized use with multi-level administrative privileges and robust authentication • Delivers robust security with digital signatures for approved patches and communications encryption

Table 1: Automated Patch Management Feature and Architectural Requirements

The BigFix Patch Manager Solution

BigFix Patch Manager specifically identifies machines that require security patches, and, under administrator control from a central console, mass-deploys appropriate patches to only those computers that require the patch. It monitors the network to ensure patches remain intact and automatically redeploys them when necessary, thus increasing network security and reducing costs. BigFix Patch Manager can manage the largest networks, supporting up to 15,000 machines from a single server.

BigFix Patch Manager delivers up-to-date patches for Microsoft, Solaris and Linux (with support for HP-UX and AIX coming in Q1, 2004). BigFix supports all Microsoft operating system and application patches released on the Microsoft TechNet or the Windows® Corporate Update Web sites, and critical patches from SUN, RedHat, and others for additional supported platforms.

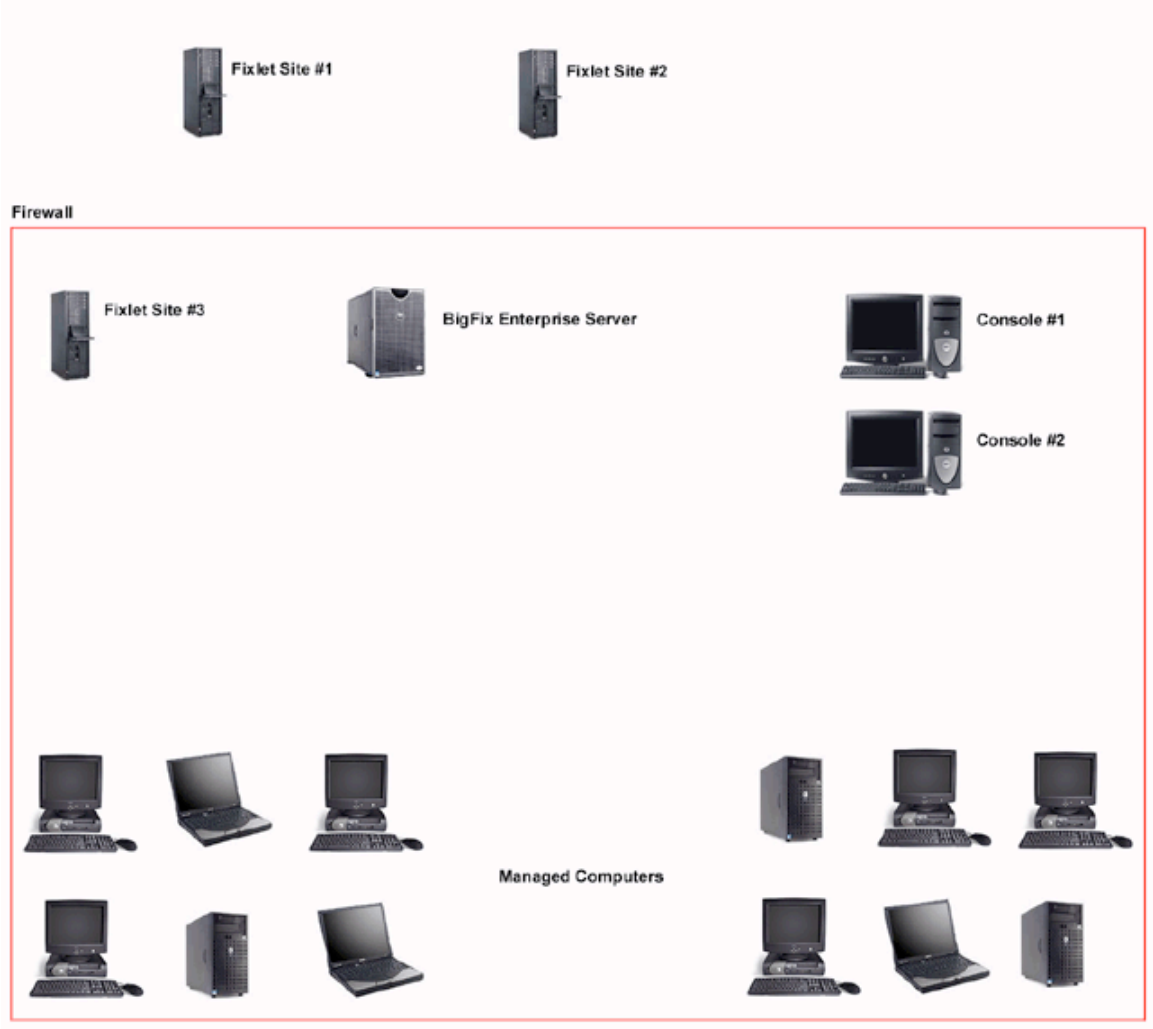


Figure 1: Major components of the BigFix Patch Manager Solution

Fixlet Creation

BigFix's patented solution centers around the Fixlet message, a small message that contains both the intelligence to detect a particular vulnerability, and the ability to fix it. Fixlet messages can be written in-house by administrators to address vulnerabilities specific to their enterprise, or can be written by external sources including BigFix and BigFix partners to address widespread problems. BigFix currently maintains multiple databases of Fixlet messages including a library of all Microsoft HotFixes, security updates, service packs, and virus-definition updates from leading anti-virus software vendors. BigFix also maintains libraries of critical operating system patches for it's growing family of supported Unix and Linux platforms. With this extensive set of Fixlet messages, BigFix Patch Manager can diagnose the state of an enterprise network, out of the box, within minutes.

A Fixlet message must be written to detect and correct vulnerability. The Fixlet message includes the intelligence to identify all instances of a particular vulnerability in the network, and provides administrators with the one or more scripted solutions that address the vulnerability on all or a subset of affected computers. Fixlet Messages are authored using the powerful BigFix Development Environment (BDE).

Fixlet Publication and Collection

After a Fixlet Message has been created and tested, it is ready to be published. This simple step publishes the Fixlet message to a Fixlet Site, making it available to all installations of BigFix Patch Manager that are subscribed to that Site. Periodically, each installation of BigFix Patch Manager that is subscribed to a Fixlet Site polls it to determine whether new Fixlet messages are available. If so, the BigFix Enterprise Server in each installation downloads the new message. Before accepting the Fixlet message, every Enterprise Server first validates that the new message is authentic by verifying the digital signature created by the Fixlet's author.

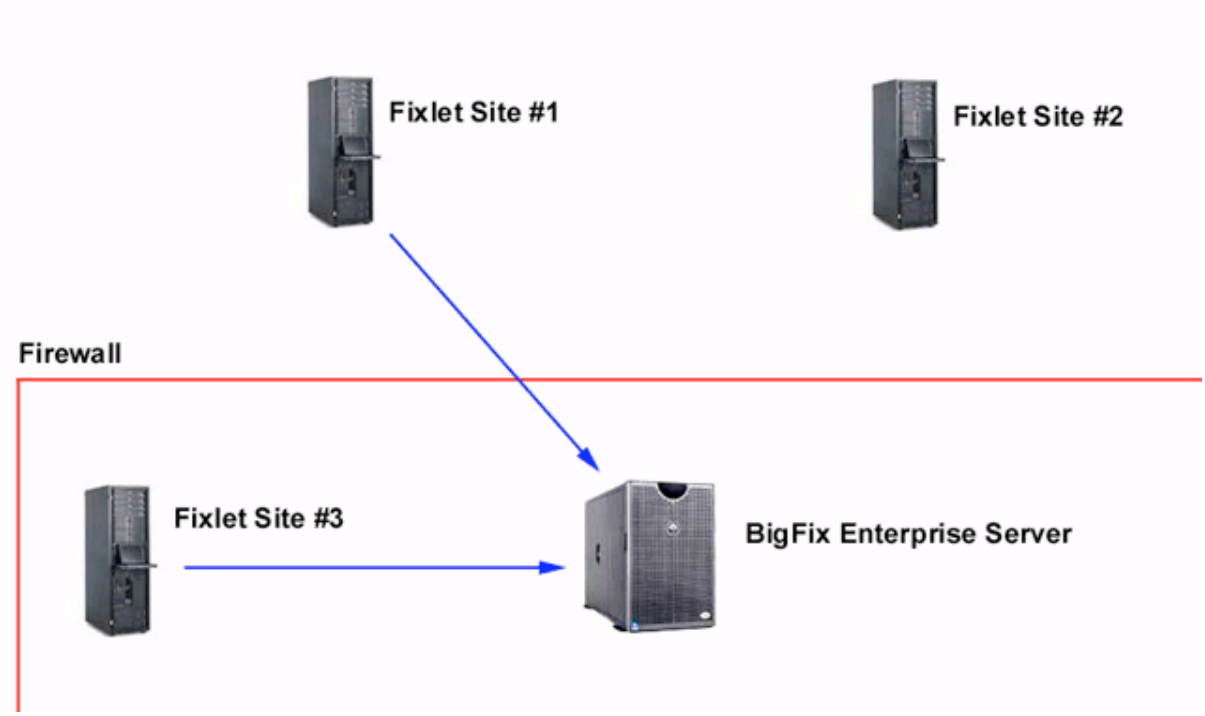


Figure 2: Fixlet collection by the Enterprise Server

Fixlet Sites may exist either outside or within the enterprise network, allowing administrators to author their own Fixlet content, or take advantage of content authored outside the enterprise.

Fixlet Distribution

As new Fixlet messages are downloaded by the BigFix Enterprise Server, the server alerts all BigFix Enterprise Agents that new Fixlet content is available, and each Agent downloads the new messages. Although the Fixlet messages can diagnose and fix very complex vulnerabilities, each one is less than 1 Kilobyte in size, ensuring that the BigFix solution creates minimal load on the network.

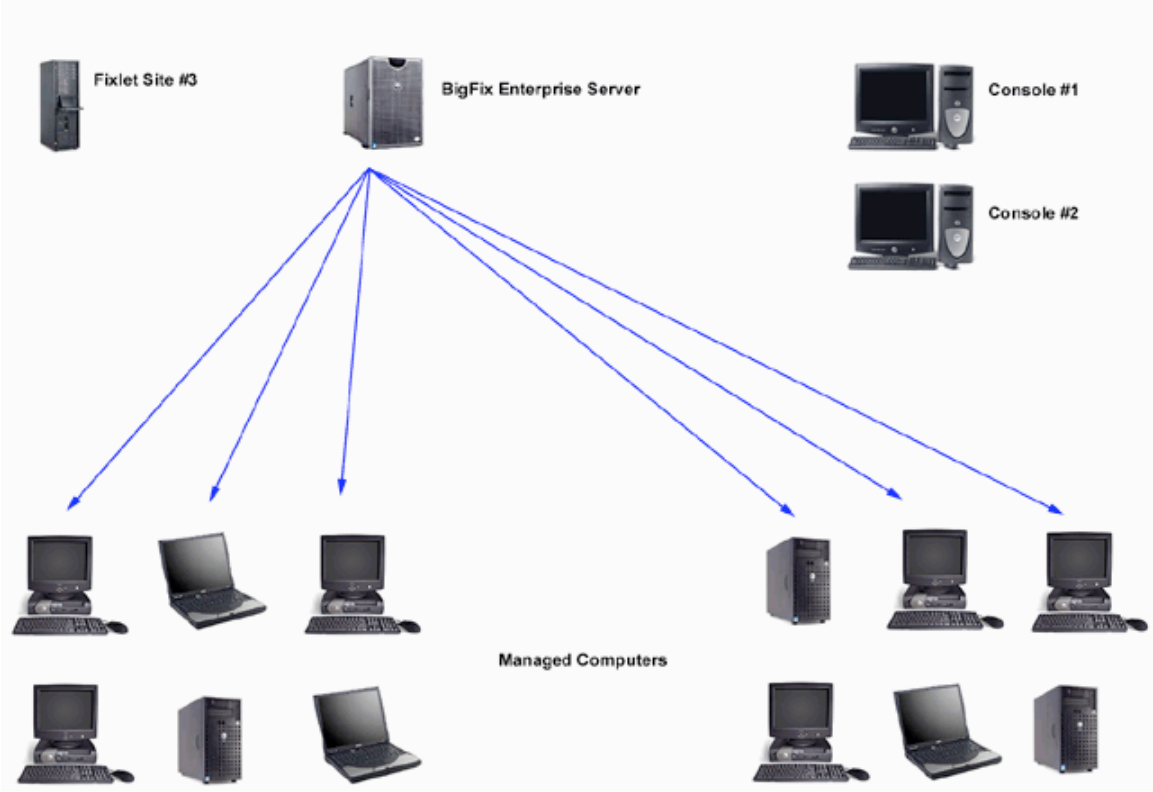


Figure 3: Fixlet distribution to the Agents

Fixlet Message Evaluation

The Fixlet message, downloaded to each Agent, contains information required for the Agent to determine whether its host computer requires that Fixlet by comparing the computer's current configuration to the conditions specified in the Fixlet message.

As an example, a Fixlet message may test for a new Microsoft vulnerability that affects only computers running Windows 2000 Server that have a particular version of a certain .dll, are running a particular service, and do not have a particular patch installed. The BigFix Agent reads these conditions from the Fixlet message and compares them to the computer's current state to determine whether the Fixlet message is required for that computer.

Relevance Reporting

The Agent reports back to the BigFix Enterprise server when it has determined that a Fixlet is required. Results from all Agents are stored in a SQL standard database on the Server, which in turn is aggregated to the BigFix Enterprise Console. The Console provides a view into all instances of the vulnerability in the network.

Note: Multiple Consoles may connect to the same server, and simultaneously administer the Agents connected to the server as needed.

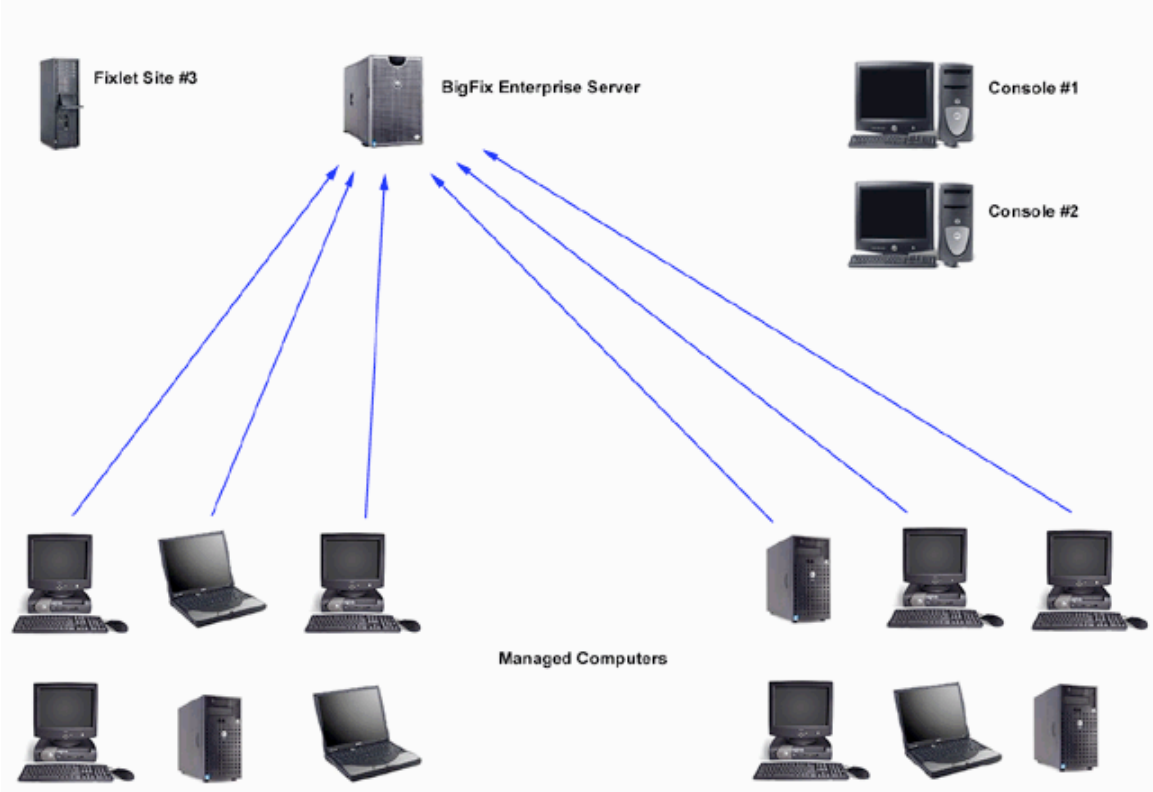


Figure 4: Relevance reported back to the Enterprise Server

Administrator Review and Decision

For each relevant vulnerability, the BigFix Enterprise Console displays the total number of computers affected, and provides the administrator with a drill down view to actual computers that exhibit each vulnerability. With this information, the administrator can deploy fixes on a case-by-case basis, or establish a policy to fix the problem throughout the network.

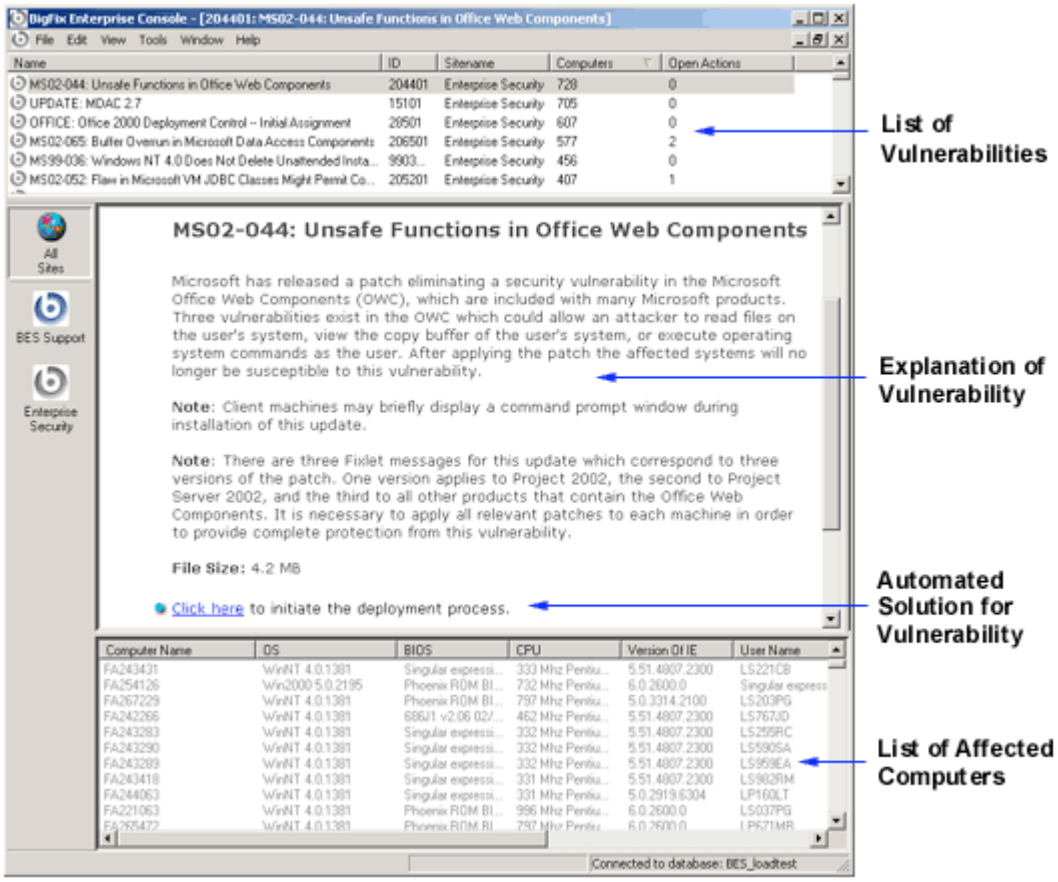


Figure 5: View of a Fixlet message in the Enterprise Console

Fixes can be applied to exactly the computers that need them with just a few clicks of the mouse. BigFix Patch Manager supports multiple deployment options including: whether a message is displayed to the user before the action is taken; whether to reboot when the action is completed; whether a reboot may be immediately executed, delayed, or aborted; and whether to deploy immediately or schedule the action to take place at a later date. The administrator can also designate a particular action as a "policy" to automatically fix any instances of the vulnerability that arise in the future without administrator action or approval.

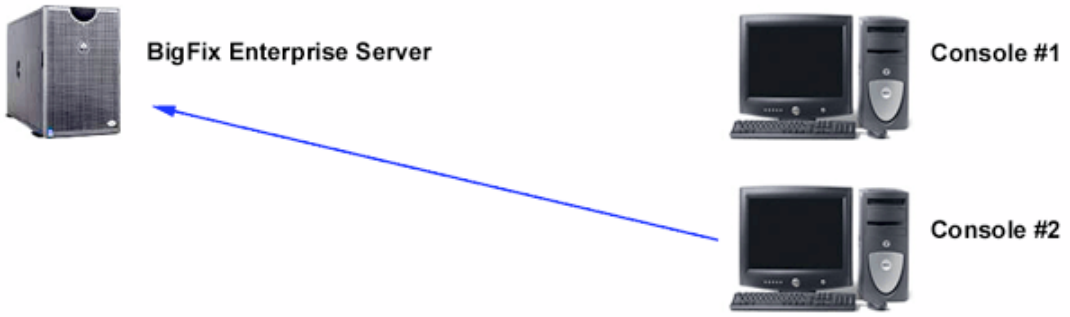


Figure 6: Action creation by Administrator

Action Distribution

Once the action has been issued, the server alerts the Agents of the action, and the agents download the action.

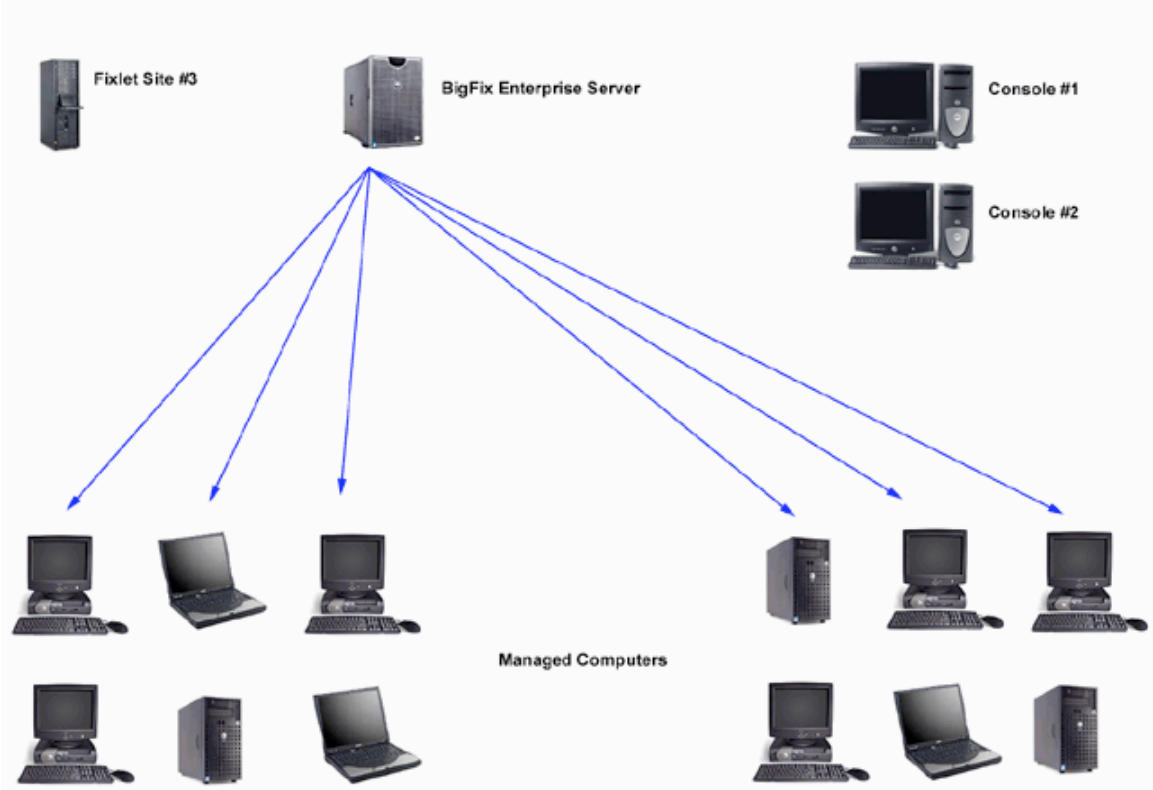


Figure 7: Action Distribution to Agents

Action Execution

After re-evaluating their state to ensure that the vulnerability still exists, the targeted Agents apply the action command. Throughout the process, the BigFix Enterprise Console displays up-to-date information on the status of deployment for all machines in the network.

Rather than have each machine download a patch or fix from the Internet, the BigFix Enterprise Server downloads a patch once from the Internet, and mirrors it to the Agents that will run the action. In bandwidth-constrained networks it is possible to further distribute these downloads using “relay servers” placed strategically within the network to further reduce network load.

State Reporting

After applying the action, each Agent reports its success or failure to the BigFix Enterprise Server, allowing the administrator to track the state of the network relative to the vulnerability. The system continues monitoring for the vulnerability even after an action has succeeded, ensuring that the administrator is notified immediately if the vulnerability reappears.

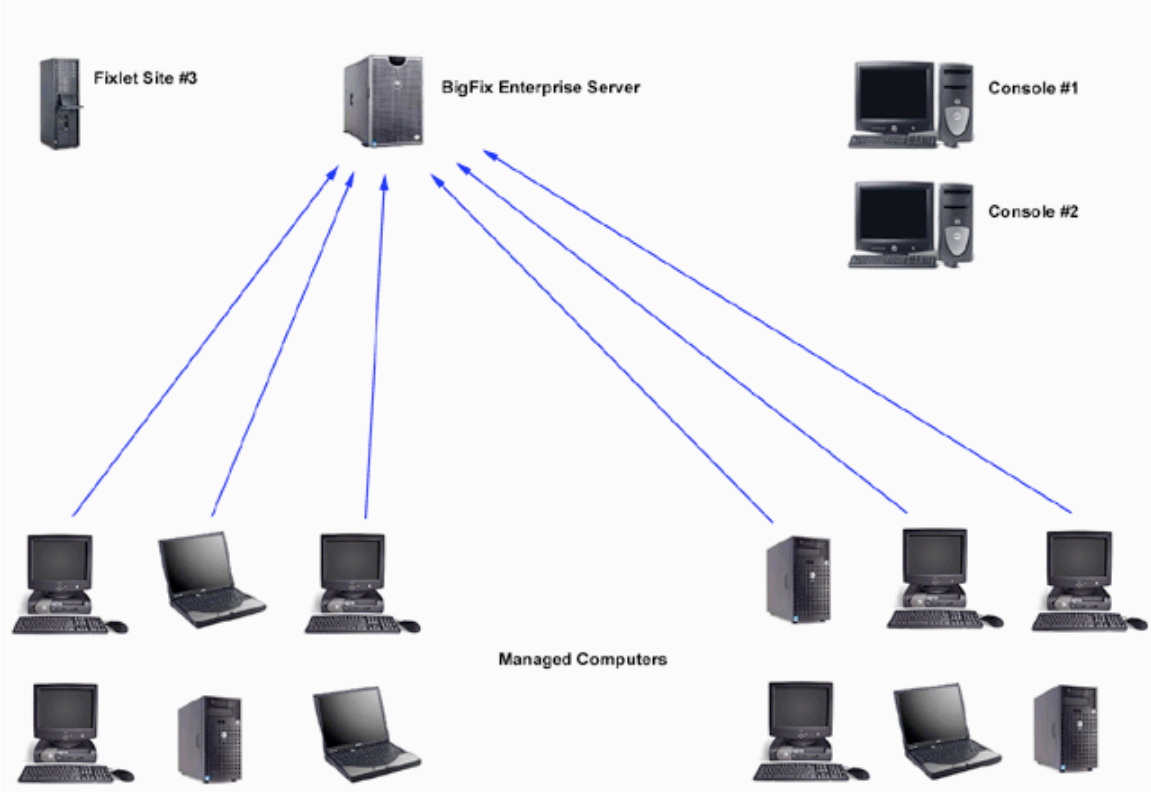


Figure 8: State Reporting by Agents to Enterprise Server

Fixlet Examples

The following examples demonstrate the types of vulnerabilities that the BigFix solution can address.

Example 1: Security Patch Deployment:
A hardware or software vendor releases a security patch, the clock starts ticking
.
The Goal: To apply the patch to all affected computers as quickly as possible, without disrupting existing services and applications.

The Process: A Fixlet message is created to detect computers that require the patch. The Fixlet message also contains all necessary code to deploy the patch to affected computers, without requiring any end-user interaction.

In the case of Microsoft Security Bulletin patches, BigFix manages a Fixlet site with a comprehensive, up-to-date set of Fixlet so the administrator need not write Fixlet messages to detect and deploy those patches. Other types of Fixlet messages can be written in-house, or by the BigFix professional service team or Fixlet content creators. Most Fixlet messages can be written in less than an hour.

Fixlet messages determine the need for the patch based on a wide variety of characteristics, including file system information (versions, sizes, modification times, hashes, contents of files, etc.), operating system information, registry values, and hardware/peripheral data. These characteristics are written into "Relevance Clauses" which are easily human readable, as in the example below:

name of operating system is "Win2000"

(csd version of operating system is "Service Pack 2") OR (csd version of operating system is "Service Pack 3")

exists file "msjava.dll" whose (version of it < "5.0.3809.0") of system folder or exists file "javart.dll" whose (version of it < "5.0.3809.0") of system folder

In addition to the Relevance Clauses, the Fixlet message also contains a textual description of the vulnerability such as including its severity rating, and an action script to download the patch and apply.

The result: The Window of Vulnerability that once measured in months can be contracted to hours, including Fixlet creation and regression testing.

Example 2: Controlling Running Processes
When Code Red hit, it affected many enterprises that should not have been vulnerable because it exploited a buffer overrun in IIS, and should only have affected servers. Yet because thousands of desktop computers were running IIS without authorization of their IT departments, these systems were compromised as well.

The Goal: To maintain control of all systems in the network to ensure that applications such as IIS are only run where they are authorized and to prohibit unauthorized applications from running on the network.

With a simple Fixlet message, BigFix allows administrators to monitor exactly which computers are running restricted applications and to determine whether they are authorized to run those applications. The following relevance clauses identify non-server computers that are running IIS:

exists registered application "inetmgr.exe"

value of setting "ComputerUse" of client does not equal "Server"

In the same Fixlet message, the administrator can then specify action required to bring these computers into compliance. In this example, the administrator could write a Fixlet message to display a pop-up dialog box requesting the user to remove the service, or automatically uninstall the application without end-user intervention.

The Result: Within minutes of authoring this Fixlet, the administrator receives a list of the non-compliant computers, and can bring them into compliance using the action script so the next Code Red will cause less damage to the network.

Example 3: User Account Settings
Password mismanagement is a hacker's dream and an administrator's nightmare. To keep the network secure, administrators must ensure that the passwords are sufficiently complex and don't become stale.

The Goal: To identify machines that are not in compliance with password policy quickly and easily.

With BigFix, this process is as simple as creating Fixlet messages to identify non-compliant accounts and fix them immediately.

A typical password policy requires that user passwords are at least 8 characters long and must be changed every two weeks. Users with administrative rights also must enable the complex

password protection in their Windows OS. Two Fixlet messages could be written to enforce these policies.

The first Fixlet message would check for users who are not administrators, and do not have an 8 character minimum password length or a password more than two weeks old. The relevance clause for this Fixlet would be:

Admin privilege of local user = False

(maximum password age of local user > 14*day) or (minimum password length of local user < 8)

The second Fixlet would detect users who were administrators, and were not in compliance. Its relevance would be written:

Admin privilege of local user = True

(maximum password age of local user > 14*day) or (minimum password length of local user < 8) or (password complexity of local user = False)

An action could then be included in each Fixlet to enforce policy. Within minutes, all user accounts that are out of compliance could be brought into compliance.

Example 4: Removing Spyware
An application such as Spyware can quickly bring a corporate network to its knees if left unchecked.

The Goal: To detect and removing Spyware.

Since Spyware doesn't register itself with Windows but always installs itself using the same system path, a simple relevance clause can find all instances of the application in the network:

Exists file "c:\program files\helpful application\spyware.exe"

The Result: Within minutes, all computers that have this application would report their state. An action could then remove all traces of the application from each computer.

Automated Vulnerability Remediation, Without the Headache

Comprehensive and easy to use, BigFix Patch Manager automates the process of patch management. As a cross-platform agent-based technology, Patch Manager distributes the intelligence to the devices attached to the network to easily and quickly identify machines that require security fixes with minimal network load. Powerful and scaleable, BigFix Patch Manger corrects problems in real-time, and lets administrators control and manage patch deployment to thousands of machines from a single server. Fast, within minutes BigFix Patch Manager can diagnose and remediate thousands of security vulnerabilities on a large enterprise network. The net result is a more secure network, with less time and money spent on patch management, and more time devoted to other high-priority network issues.