OpenBSD: Security for Financial Environments

This article discusses how the use of a free, open source operating system can be used to increase security and reduce costs.

BY NICHOLAS C. P. HUMPHREY AND PETER WILD

Royal Holloway University of London

INTRODUCTION

Many organisations contain a highly sensitive financial hub where top-tier control is kept over payments, accounts, financial transaction handling and reporting. Whether a large insurance firm or regional bank, this central Treasury will be a crucial part of the organisation, and the confidentiality, integrity and availability of data stored there may mean the difference between business success and failure. In modern business, the centralised financial control model is becoming less well defined, with many responsible personnel and sensitive functions being spread across diverse parts of the organisation. With branch offices and remote workers, organisations face new challenges of providing secure communications as traditional boundaries begin to blur. A layered, carefully considered approach to security across the organisation will help address the wide range of electronic and non-electronic threats which are present.

OpenBSD is computer software developed by security enthusiasts in their spare time and made freely available (with all source code) to the worldwide community under a relaxed licence. OpenBSD, like other computer operating systems such as Solaris, Windows and HP-UX is loaded onto computer hardware and acts as the base on which business applications (for example, electronic funds transfer software) are run.

OpenBSD: Security for Financial Environments

Nicholas C. P. Humphrey
Information Security Group,
Royal Holloway, University of London
Egham, Surrey, U.K.

Peter Wild
Information Security Group,
Royal Holloway, University of London
Egham, Surrey, U.K.

This article was prepared by students and staff involved with the award-winning M.Sc. in Information Security offered by the Information Security Group at Royal Holloway, University of London. The student was judged to have produced an outstanding M.Sc. thesis on a business-related topic. The full thesis is available as a technical report on the Royal Holloway website http://www.ma.rhul.ac.uk/tech.

For more information about the Information Security Group at Royal Holloway or on the M.Sc. in Information Security, please visit http://www.isg.rhul.ac.uk.

OPENBSD SECURITY

Modern organisations are often subject to numerous statutory, regulatory and industry compliance requirements (for example, the Payment Card Industry Data Security Standard and the Data Protection Act). Such requirements often involve the need to demonstrate that systems are secure against attacks on confidentiality, integrity and availability. OpenBSD's audited code base has a reputation amongst security circles for having fixed many issues long before they

become problematic in other operating systems. An example of this is where the OpenBSD developers had changed an algorithm in their version of the BIND DNS server code, avoiding a recently published DNS cache poisoning attack [1]. OpenBSD's security features provide numerous benefits over other operating systems, by starting with a minimal configuration which is added to as necessary, rather than defaulting to activating all functionality which then requires systems administrators to remove the parts which are not needed. Additionally, OpenBSD's built-in network and application controls allow the operating system to define fine-grained controls over what business applications and users may do, increasing confidence that systems are only performing actions which are explicitly allowed by organisational policy.

OpenBSD contains a number of technologies which can be applied to financial environments to create a layered approach to security. From IPSec (IP Security) secured network connections, to secure remote administration over OpenSSH (Secure Shell, a replacement for Telnet). OpenBSD also features a free, built-in firewall which is highly configurable and can perform filtering on both the perimeter of the network filter and also on the host itself. The firewall can limit network communications to only those approved systems and networks with which the system should be able to exchange data. The "pf" firewall is configurable to restrict both inbound and outbound traffic from the system, allowing a policy to be defined over not just what comes into the system, but also what leaves it.

Since version 3.8 of OpenBSD, the swap file (virtual memory which allows the use of hard disk storage, also known as paging) is encrypted by default. This means that data which is left in swap file on the hard disk when the system shuts down cannot be recovered without the correct key. Such functionality has only recently become available as an included feature in Microsoft Windows Vista, and even then only in certain enterprise versions. Swap files can potentially include extremely sensitive data, including full details of financial transactions being processed by software on the host.

OpenBSD contains numerous memory protection features to reduce the risks associated with poorly coded or malicious programs. Most of these features within

OpenBSD contains a number of technologies which can be applied to financial environments to create a layered approach to security.

OpenBSD could be viewed as an indirect (and in some cases, transparent) benefit to using the operating system. With some of the key controls based within the kernel, the benefits are derived simply from using applications on the operating system and without any user or administrator involvement. With regards to security within a financial environment, these additional controls could be regarded as compensating or mitigating controls for potentially problematic applications. OpenBSD's ability to emulate other UNIX systems (such as HP-UX) means that an application could be hosted on OpenBSD with binary emulation, and with security features such as memory protection, the "pf" firewall and Systrace, allowing for a much tighter control over the execution of that business application. In using OpenBSD to host financially sensitive applications, auditors could be given a higher degree of confidence over the execution security of those applications, as they would be subject to constraints which make exploitation by attackers that much more difficult, such as the injection of shell code [2, page 84], controls which may not be available on the application's native Operating System.

LOCKING DOWN BUSINESS APPLICATIONS

One of the more flexible tools in the OpenBSD system is Systrace, which has been included in the base system for several years. Systrace "monitors and controls an application's access to the system by enforcing access policies for system calls" [3]. This allows system administrators to define a security policy for each individual application executed on the system, from reading and writing files to controlling whether the code may make a network connection and, if so, to what. These policy elements can include fine grained control over execution context of the code, for example only allowing applications to perform sensitive functions if executed by a user with the appropriate group membership. Such control even supersedes the all powerful 'root' (UID 0) user, which traditionally has complete control within a UNIX environment. By generating, fine tuning and enforcing Systrace policies, additional constraints can be applied to applications above and beyond the level of control afforded by traditional UNIX control mechanisms such as permission masks and group membership. A good example of Systrace's flexibility can be demonstrated in its ability to constrain an

One of the more flexible tools in the OpenBSD system is Systrace, which has been included in the base system for several years.

application to only be able to read and write to certain files, whilst preventing that application from sending or receiving any network traffic whatsoever. Such a set of controls may be extremely useful in providing additional security barriers for third party software, some of which may be emulated, closed-source binaries which cannot be directly audited as their code is not publicly available for review. This means that an application can be given only the privileges it needs to get the job done (Least Privilege). In a financial or other sensitive environment, such control can be split between numerous applications and users, allowing for separation of duty and more robust auditing.

Although Systrace provides many advantages, it should not be considered a magical solution to application security, as it is not without its own security faults. In late 2007, a major flaw was discovered in Systrace, allowing an attacker to escalate his privileges and bypass auditing[4]. The authors of the popular SysJail virtualisation environment consider this flaw to be so severe that they have placed a notice on the front page of their website urging all users of SysJail and Systrace to discontinue using the tools until further notice[5]. Thus care must be taken in the use of Systrace.

OPENBSD IN ACTION: SECURING ACCESS TO LEGACY APPLICATIONS

In modern business one of the frequent challenges is being able to use existing legacy systems to deliver capabilities to clients, partners and suppliers. Consider a scenario in which an organisation has a mainframe or esoteric UNIX application sitting within the core network, but the application can only be accessed by locally attached serial terminals or by insecure telnet connections. This of course presents security issues and access problems for those outside the organisation, and there are several approaches to resolving this. A low-cost way of providing secure access to the application is by utilising an OpenBSD system to provide secure access to both internal and external users. The OpenBSD server ("Bridgehead") can be wired to one or more serial lines on the legacy system ("Target"). Target is configured to only accept connections originating from Bridgehead. Bridgehead can then be configured to accept remote users in a number of ways, from attached modems or accepting users over the network. Users would connect to

Although Systrace provides many advantages, it should not be considered a magical solution to application security, as it is not without its own security faults.

Bridgehead and use it as a jump off point to connect directly to Target, removing the need for expensive “screen scraping” applications or complex middleware to broker exchanges. The Bridgehead could accept users over an SSH connection, providing encrypted communications between the user’s terminal and Bridgehead, and perform the last hop to Target over simple, standard serial lines. The user’s access to the serial terminal on Target could be automated and simplified with appropriate shell scripting, presenting transparent access to users. OpenSSH could be configured to use public key authentication, removing traditional password based authentication and making a brute force attack against Bridgehead almost impossible. More complex configurations could utilise OpenBSD’s built-in IPSec (IP Security) implementation, allowing for the creation of a VPN (Virtual Private Network) connection between the user and Bridgehead. In addition, the ”pf” firewall can be configured to restrict incoming IPSec or SSH connections to those originating from specific individual or groups of approved IP addresses. The OpenBSD system could be configured with additional verbose session logging to a remote hardened syslog server or even output all sessions to a physical line printer, providing a trail of evidence which would be extremely difficult to dispute in the event of misuse or unauthorised access.

OPENBSD OR PRODUCT X?

One of the core considerations for any security implementation is evaluating the quality of products intended for deployment. If a single technology is chosen to provide security across multiple layers of the architecture, then a failing in that product may affect all layers on which it is installed. For example, in highly sensitive environments employing two or more firewalls in serial to filter traffic, if the same firewall product is used, then a flaw found in the outer most perimeter would also be present on the inner perimeter, making an attacker’s job a case of simply exploiting the same flaw twice to gain the required access. By diversifying the product range used in such scenarios, a flaw in one product may not be present in others, requiring an external attacker to penetrate two layers of firewall defence before being able to breach the inner perimeter. A diverse environment brings with it other challenges and considerations, notably the need to train support staff in both product lines, to patch

One of the core considerations for any security implementation is evaluating the quality of products intended for deployment.

and to maintain two sets of systems and also to maintain credentials for both systems unless a single sign-on is supported. We remark that single sign-on itself may lead to issues where one compromised set of credentials could then be used to alter security devices to the attacker's desired configuration. Such consideration is technology neutral, and requires careful thought. An organisation must perform an adequate risk assessment to explore such issues, determine their appetite for risk and the available budget. Wider organisational issues or internal politics may also drive a particular technology's usage, but all developments of hardware and software should be carefully considered as to their impact on the platform as a whole.

OPENBSD: THE BIG PICTURE

OpenBSD, like any software product, is not without its own issues and as such should not be considered as the ultimate answer to security. Specific technologies may play a part in attitudes towards the overall management of security in the enterprise, but systems and technologies are only one part of the security puzzle. The creation and maintenance of suitable policies, procedures, processes, guidelines and standards should be undertaken in concert with proper risk management across the enterprise, aligned with external obligations, business objectives and the capabilities of the corporate assets - infrastructure, personnel and business processes. OpenBSD can be used to increase the level of assurance in certain environments, provided that the application of the technology is done in cooperation with a security strategy which is driven by management support and effectively communicated to staff.

Ron Condon

UK bureau chief
searchsecurity.co.UK

Ron Condon has been writing about developments in the IT industry for more than 30 years. In that time, he has charted the evolution from big mainframes, to minicomputers and PCs in the 1980s, and the rise of the Internet over the last decade or so. In recent years he has specialized in information security. He has edited daily, weekly and monthly publications, and has written for national and regional newspapers, in Europe and the U.S.

REFERENCES

[1] Schlyter, J. OpenBSD & BIND 9 cache poisoning - http://marc.info/?l=openbsd-misc&m=118539211412877&w=2 (Accessed 2007-07-28)

[2] Erickson, J. M - Hacking: The Art of Exploitation - No Starch Press, 2003.

[3] OpenBSD - man page: systrace(1) - http://www.openbsd.org/cgi-bin/man.cgi?query=systrace\ &apropos=0\&sektion=1 \&manpath=OpenBSD+4.1\&arch=i386\ &format=html (Accessed 2007-07-30)

[4] Watson, R. N. M - Exploiting Concurrency Vulnerabilities in System Call Wrappers - http://www.watson.org/~robert/2007woot/2007usenixwoot-exploitingconcurrency.pdf (Accessed 2007-08-13)

[5] Sysjail Project - Sysjail: A Userland Virtualisation System - http://sysjail.bsd.lv/ (Accessed 2007-08-12)