

- Computers & Security, vol. 21, no 1, pp62-73.
- 12 Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C.E. 1996. "Role-based access control models", IEEE Computer, 29(2), pp38-47.
- 13 Chung, C.Y., Gertz, M. and Levitt, K. 1999. "DEMIDS: A Misuse Detection System for Database Systems", in Proceedings of the 3rd International Working Conference on Integrity and Internal control in Information Systems, Amsterdam, The Netherlands, 18-19 November 1999.
- 14 Low, W.L., Lee, J. and Teoh, P. 2002. "DIDAFIT: Detecting Intrusions in Databases Through Fingerprinting Transactions". in Proceedings of the 4th International Conference on Enterprise Information Systems, Ciudad Real, Spain, 2-6 April 2002, pp121-128.
- 15 Phyto, A.H., Furnell, S. and Ifeakor, E. 2004. "A framework for monitoring insider misuse of IT applications", in Proceedings of Information Security South Africa 2004, Johannesburg, South Africa, 30 June - 2 July 2004.
- 16 BSI. 2001. Information technology. Code of practice for information security management. BS ISO/IEC 17799:2000. British Standards Institution, 15 February 2001. ISBN 0 580 36958 7

About the author

Dr Steven Furnell is the head of the Network Research Group at the University of Plymouth, UK. He has been actively involved in security research for over 12 years, and has authored numerous papers on the topic, as well as the book 'Cybercrime: Vandalizing the Information Society', published by Addison Wesley. He is currently leading a project in relation to insider misuse detection, funded by the UK Engineering and Physical Sciences Research Council.

Contact

sfurnell@network-research-group.org

Cisco source code stolen – but should we care?

Philip Hunter

It was clearly bad news for Cisco itself when a portion of its IOS software surfaced for a few days in May on a Russian website. But it was difficult to obtain a consensus within the security industry over the potential threat posed by the breach to the Internet as a whole or to the countless private IP networks. Given that IOS drives most of the world's routers that direct traffic both through the Internet and private networks, theft of some of its source code clearly gives hackers the potential to exploit vulnerabilities that would be hard to identify otherwise. Naturally the Open Source community pounced on the issue, as they did earlier in the year when some Microsoft Windows source code was stolen, with the argument that any system relying on secrecy for security is fundamentally flawed and by definition insecure. Kerckhoff's law that "a system should be designed to be secure if everything is known about it except the key information" was trotted out as an argument that closed source software such as IOS and Windows would soon be extinct, ushering in the golden age of open source.

Secrecy for security's sake?

In truth this argument is rather disingenuous. Source code may be kept secret for commercial reasons, and doing so does not automatically imply that it is less secure. The test is whether the vendor is relying significantly on secrecy for security, in which case it is untrustworthy. But in the case of Cisco and Microsoft the sit-

uation is slightly ambiguous in that while intellectual property protection is the primary motive for source code secrecy, both vendors have relied to some extent on obscurity for their security. At any rate that has been the practical outcome of their efforts to maintain source code secrecy, for exposure even of portions of code usually leads to the discovery of previously unknown vulnerabilities.

Is the leak a threat?

The question is whether the newly exposed vulnerabilities pose a serious threat. In the case of Cisco it is too early to tell, although there are some pointers. In the case of Microsoft, exposure of the source code revealed a hole in Internet Explorer (IE), but this has yet to be exploited, and no further vulnerabilities have been discovered. Hackers continue to find ways of attacking Windows and IE without having to probe the source code.

The theft reinforced...the futility of keeping source code secret

That being the case, Cisco should be in a still more comfortable position, at least with regard to the source code exposure. Hackers have hitherto concentrated most of their fire on the servers, hosts and private networks attached to the Internet, rather than on the underlying IP infrastructure. Whether the motive is notoriety, fraud or disruption, end systems constitute a more rewarding target.

On the other hand a successful large scale attack on the Internet itself could

create more widespread economic damage and undermine confidence in online activity as a whole, so the Cisco breach has ramifications beyond the company's own credibility.

However, even Cisco's own credibility cannot be entirely dissociated from the well being of the Internet, given the company's dominance of the router market. If its routers cannot be trusted, neither can the Internet itself. Furthermore a number of service providers have become customers of Cisco's security products and many will have been eagerly awaiting the company's recent announcements of Network Admission Controls Solutions (NACS), designed to give private networks greater protection against viruses and worms. Cisco is hoping that third party security vendors support this software within their anti-virus and other products, so that it becomes as ubiquitous a platform for defending against intrusions as IOS itself is for IP networks as a whole. NACS comprises distributed agent software that monitors the status of security components such as anti-virus software from various clients and servers on a network.

There is no question at present of vendors failing to endorse NACS because of the IOS security breach, and both Symantec and Network Associates have already made commitments. But some customers, including Internet Service Providers, have expressed private concerns, and are holding back on decisions to deploy Cisco security software until the investigation into the breach has been concluded.

Was it Cisco's fault?

Since the FBI confirmed its involvement in the investigation in late May, Cisco has declined to make any meaningful comment, and this was still the position at time of writing. However the company did admit that customers might link the incident to judgement of Cisco's own security software if it turned out the breach resulted from an internal security lapse. "There are various ways it could have happened, but we can't comment on the manner in

which it was done until the investigation is complete," said a spokesman. This, the spokesman admitted, could take several more months.

Again the security community was divided over the source of the leak. Early evidence implicated a Sun Sparc server as the source, accessed via a Virtual Private Network (VPN) connection, leading some to conclude that it must have been within Cisco's internal network. This was based on the assumption that other parties with access to IOS source code such as software developers or remote workers would be more likely to have the software on a laptop, and come in over a or dial up connection and not VPN.

But even as circumstantial evidence this is dubious, for there are plenty of third party developers with access to IOS source code on servers. Until the server's identity is obtained it is impossible to draw any firm conclusions merely from its type.

Minor risk

When it comes to the risk, there are reasons for cautious optimism. As already noted hackers tend to be less interested in attacking routers, or have been until now. And while the amount of code claimed to have been stolen from Cisco, about 800 MB, is considerably more than in the case of Microsoft, the perpetrators have only revealed a small proportion of it. Furthermore the Microsoft code was freely available in a clean state ready for download and was reasonably easy to work with. This was not the case for the Cisco code, which would require in depth knowledge of the router hardware and of IP routing to make any sense of. At present it is believed that few hackers would have the expertise to pore through 800 MB of IOS source code and identify new vulnerabilities to exploit. The exception would be hackers who have worked either for Cisco or in the IOS developer community. Such people may exist, but they might well have legitimate access to IOS source code anyway. The conclusion therefore is that the theft reinforced the similar Microsoft incident in highlighting the futility of keeping

source code secret, but does little to add to the existing burden of risk.

Some experts suggest that access to IOS source code could help hackers mount large scaled distributed denial-of-service attacks against Cisco routers. This possibility exists, but unless Cisco has been asleep for the last 10 years it will have engineered IOS to withstand DOS attacks.

The greater potential threat is that if a sufficient amount of source code were available to hackers, they could modify Cisco's licensing mechanisms, enabling them to create illegal copies incorporating backdoors or Trojan Horses that could be exploited subsequently to gain entry. At present though it is not known which parts of the source code were leaked, for neither Cisco nor the attackers have revealed it. This is thought to be a bigger danger than an immediate worm, posing a rather longer-term subliminal threat that is hard to quantify.

The real question perhaps is whether such incidents in which a portion of source code of a major software platform on which almost every business depends is stolen in some way really increases risk of attack. Clearly they do bring some risk increase, but there is little evidence so far that there have been more security breaches as a result. Indeed few if any of the major attacks have relied on access to source code, although they have exploited vulnerabilities, for example in Windows, resulting from failure to make security a sufficiently high priority during successive software design cycles.

It is true that by making software open source, the impact of code theft is entirely eliminated. But as has been argued before in this newsletter, making software open source does not prevent there being unknown vulnerabilities. These vulnerabilities may not be hidden in the sense of being shielded from public view, but are unknown because the code is so complex and convoluted that no one has yet discovered them. It may be that open source software will ultimately prove to be more secure because of its widespread scrutiny and the fact that it does not rely on secrecy for security, but that case has yet to be finally proven.