

University of Jordan and is interested in research in the fields of wireless sensor networks and clustering techniques.

Dr Ahmad Sharieh was dean of Sur University College, Sur, Sultanate of Oman and dean of King Abdullah II School for Information Technology, University of Jordan. During 2005-2007 he worked as a director of development and quality assurance affairs at The University of Jordan. He was chairman of the computer science department, Amman Arab University for Graduate Studies, Jordan and served as chairman of the Central Tenders Committee. Previously he was co-ordinator of the University Procurement Unit of the Higher Education and EUROP Banking Projects, Jordan and assistant dean of the Academic Research Deanship, University of Jordan.

Dr Azzam Sliet is the former Minister of Information and Communications Technology (2013-2015). He is currently working as a professor of computer science, King Abdullah II School for Information Technology, University of Jordan (www.ju.edu.jo), where he functioned as the dean (2015-2016) and the assistant president/director of the computer centre (2007-2009).

Nidaa Al-Azzam is an instructor at the computer science and information technology school at Al-Dammam university in Saudi Arabia.

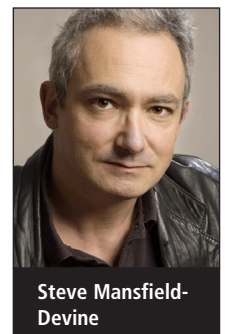
References

1. Boncella, Robert. 'Wireless Security: An Overview'. Communications of the Association for Information Systems, Vol.9, 2002, pp.269–
2. Shi, E; Perrig, A. 'Designing Secure Sensor Networks'. IEEE Wireless Communications Magazine, vol.11, no.6, Dec 2004, pp.38–43. Accessed Jul 2017. <http://ieeexplore.ieee.org/abstract/document/1368895/>.
3. Yong, Wang; Attebury Garhan; Ramamurthy Byrav. 'A Survey of Security Issues In Wireless Sensor Networks'. CSE Journal Articles, Paper 84, 2006. Accessed Jul 2017. <http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1087&context=csearticles>.
4. Rivest, R; Shamir, A; Adleman, L. 'A Method for Obtaining Digital Signatures and Public-Key Cryptosystems'. Commun. ACM, vol.26, no.1, 1983, pp.96–99. Accessed Jul 2017. www.dtic.mil/get-tr-doc/pdf?AD=ADA606588.
5. Gura N; Patel A; Wander A; Eberle H; Shantz SC. 'Comparing elliptic curve cryptography and RSA on 8-bit CPUs'. In CHES (Vol.4, pp.119–132), Aug 2004. Accessed Jul 2017. <https://pdfs.semanticscholar.org/17eb/de1ba63ade72a0419bfee05c4fb34ae37aa0.pdf>.
6. Wander AS; Gura N; Eberle H; Gupta V; Shantz SC. 'Energy analysis of public-key cryptography for wireless sensor networks'. In Pervasive Computing and Communications, PerCom 2005. Third IEEE International Conference 8 (pp.324-328), Mar 2005. Accessed Jul 2017. <http://dl.acm.org/citation.cfm?id=1049786>.
7. Liu, Y; Ning, P; Dai, H; Liu, A. 'Randomized Differential DSSS: Jamming-Resistant Wireless Broadcast Communication'. Proceedings IEEE INFOCOM, pp.1–9, Mar 2010. Accessed Jul 2017. <http://ieeexplore.ieee.org/document/5462156/>.
8. Strohmeier, Martin; Lenders, Vincent; Martinovic, Ivan. 'On the Security of the Automatic Dependent Surveillance-Broadcast Protocol'. IEEE Communications Surveys & Tutorials, 2014. Accessed Jul 2017. <https://ieeexplore.ieee.org/iel7/9739/5451756/06940209.pdf>.
9. Saleem, K; Faisal, N; Abdullah, M; Hafizah, S. 'Biological Inspired Secure Autonomous Routing Mechanism for Wireless Sensor Networks'. International Journal of Intelligent Information and Database Systems (IJIIDS), 2010. Accessed Jul 2017.
10. Saleem, K; Khalil, M; Faisal, N; Ahmed, A; Orgun, M. 'Efficient Random Key based Encryption System for Data Packet Confidentiality in WSNs'. 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013. Accessed Jul 2017. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6681032>.

Open source and the Internet of Things

Steve Mansfield-Devine, editor, Network Security

The Internet of Things (IoT) is gaining a reputation for insecurity. Researchers have revealed dangerous flaws in everything from baby monitors to power stations. And much of the problem stems from an attitude of 'innovate now, secure later'. In this interview with Tim Mackey, technical evangelist at Black Duck Software by Synopsys, we look at what the IoT encompasses, why so much of it appears to be flaky or downright dangerous and how the unquestioning and promiscuous use of readymade solutions, including open source code libraries, can have serious consequences.



Steve Mansfield-Devine

What is IoT?

Perhaps the most immediate problem with the IoT, however, is deciding what that term covers. If you take the word of pundits and vendors it seems to encompass everything from baby monitors to



Tim Mackey is technical evangelist for Black Duck Software by Synopsys. He engages with technical communities to learn what bleeding edge security concerns are top of mind in order to feed them back into the development team. He is well versed in open source application security, datacentre security, containers, virtualisation and cloud technologies. Mackey has spoken at many events, including OSCON, CloudOpen, Interop, CA World, Cloud Connect and the CloudStack Collaboration Conference. He is also a published O'Reilly Media author.

the supervisory control and data acquisition (Scada) systems running electricity grids and factories. Indeed, Mackey's take on how that label can be applied is quite liberal.

"I tend to go for simple," he says. "It is a computing device that is connected to a network that is publicly accessible. And 'publicly accessible' could be within a business entity – so that would encompass Scada – if there is a mechanism by which someone on a public Internet could gain access to a terminal that is then connected to the Scada network, and then go and do whatever they want to do. That would be an Internet of Things scenario for me. I tend to avoid the consumer label on it, as it tends to have fearmongering associated with it – attacks of doorbells and microwave ovens and fridges, and so forth."

However, that raises another question. Systems have been publicly accessible via Internet-facing interfaces for some time. So why are we witnessing so much discussion right now?

"Popularity," says Mackey. "It becomes more front of mind for people because they are now dealing with connected devices. For Christmas I got an Alexa Dot and have to figure out how to make that actually be useful without spying on what

my children say during the day, as interesting and intriguing as that might be. But the reality is that these devices have been around for 20 years."

He harks back to a time, early in his career, when he was connecting human-managed systems in a factory to a centralised management system using ladder logic.¹ The aim was to enable just-in-time inventory management and other efficiencies. But as with many IoT projects, this often involved adding interfaces to systems that were never originally designed to have them.

"That was the model 20 years ago," Mackey says. "We've just extended it dramatically, connected it to the public Internet and then, with the rise of cybercrime, we've looked at it through a risk-profile lens." The image this lens shows us, he says, is that, "there's a lot of scary out there – what do we need to do next?"

And that's as far as we've got, says Mackey. "On the industrial side of things, there's tacit recognition that how we've been securing these devices and trying to run facilities with a minimum staffing – largely a non-technical staffing at that – presents a real challenge. And then you factor in all of the consumer-grade devices that are flowing through Amazon's shelves, all the way to things that are truly odd, like an Internet-enabled Barbie dream house."

What's the problem?

There is also an issue of scale here, too. Internet interfaces are nothing new, but the sheer number of devices with IP addresses seems to be ramping up exponentially. Vendors, it seems, are attempting to differentiate their products by throwing in a web interface or an Internet-based back-end service.

"One of the things that I've tried to impress upon the engineering teams I've worked with over the years is that just because you can, doesn't mean you should," says Mackey. "Do I really need a Bluetooth-enabled toothbrush?"

The cost of adding Bluetooth, wifi or other network interfaces to devices is now trivial. And there are readymade solutions that are very simple to implement, at least from an engineering

standpoint. But as Mackey points out, the process doesn't stop there. Vendors need to ask themselves if they really know how to secure those interfaces.

"Is the security of the connection a core competency of the vendor who's supplying it?" he asks. "That starts to focus the problem around prioritisation of effort. We know, in tech, that securing things is an ever-increasing problem and, by extension, requires a level of competency that isn't there – especially if you want to effectively do a land grab around the piece of technology or a particular capability."

By 'land grab', Mackey means being first to market – grabbing as large a slice of the market as you can before your competitors weigh in. It's that sense of urgency that causes security to be forgotten. And while that might be an immediate issue, at some point it can come back to bite you.

"History has shown us, pretty much since the beginning of computing time, that there is always some software defect that someone will be able to exploit eventually," he says. "It may take years to surface, but it's in there."

What you do about that is a tricky question. Patching operating systems and applications on desktop and laptop machines is well understood and the mechanisms for doing it are highly advanced – but even so it works imperfectly with many machines running unpatched and vulnerable software. But what about when that code is held in firmware with no over-the-air (OTA) updating mechanisms to support it? And there's also uncertainty, as Mackey points out, about what we should regard as the lifecycle of these devices.

"I was truly shocked to hear, a few years ago, that some of the original devices that I'd put out when working for a company 15 years ago were still very much in service," he says. "Those were industrial systems and they had an expectation of serviceability that measured in a decade plus. I don't believe the Barbie dream house will have such a longevity to it, but nonetheless, the vendors aren't necessarily applying the same level of ownership to the software. There's a certain expectation that consumers might

have to deal with this, and the way that they deal with it is by just upgrading to the next device. We saw that with the Mirai bot, having an OpenSSH port and being able to accept commands, and so becoming part of a botnet. That kind of nonsense should not exist.”

The Mirai botnet was built mainly through compromising digital video recorders (DVRs) used in CCTV systems.² This is precisely the kind of product where little thought is given to providing a means of updating it if a vulnerability is discovered and it's thought that many of the devices exploited by Mirai to mount attacks such as those on security journalist Brian Krebs and the Dyn DNS service are still just as vulnerable and still connect to the Internet.

“Their owners are, in all likelihood, blissfully unaware that their DVR is in fact compromised and is part of a greater malicious act,” says Mackey. But he also uses this as an example of how difficult these devices are to secure. Let's say, for example, that the manufacturers of the devices that were compromised had done their due diligence and had taken all the necessary steps to ensure their products were secure. They might have performed a security review that presented a perfectly good rationale for having an OpenSSH implementation running in the devices. The presence of such a service would normally ring alarm bells (especially when coupled with default passwords). And an examination of known vulnerabilities in the Common Vulnerabilities and Exposures (CVE) database, managed by Mitre, would have shown an issue.³ But this CVE dates from 2004.

“If you read the description, it describes a world that doesn't exist anymore,” says Mackey. “It quite literally refers to services that have been turned off for years. So it's completely rational to conclude, well, this doesn't apply to us any longer. It requires going one level deeper to find out that there was, in fact, a default configuration that could prove unfortunate in the event of default passwords being present. And that's really what the story turned out to be – that a set of well-known passwords were in play on an Internet-connected device.”

Ownership and responsibility

With no OTA updates available for devices such as DVRs, what are our options? Do we have to just put up with vulnerable devices until they break or are updated with a better (and hopefully secure) model?

“It's an ownership and responsibility problem,” says Mackey. “The vendors need to have some mechanism by which they can recognise that these devices are connected to accounts, and notify the users through their accounts. Forget filling out registration cards and putting them in the post and hoping for someone to actually give you a recall notice. There needs to be a level of interactivity.”

“There's a certain expectation that consumers might have to deal with this, and the way that they deal with it is by just upgrading to the next device”

In the case of a DVR, for example, it might come with a mobile app used to view the video on the device. This would also offer a mechanism for interacting with the vendor, which can then push out notifications about security issues and any potential fixes. In many cases, this could even be automated to an extent. “It might prompt you for a new password, so that you're not using a default password,” says Mackey, “or turn off some network services that probably ought not to have been on in the first place. Maybe it gives an opportunity to download new firmware.”

Another way of saying this is to recognise that there's a human in the loop at some point. Most IoT devices have that Internet connection in the first place to offer communication between device and user. Often there's a web-based service involved too. This offers the architecture needed to at least communicate problems to users.

Buying in

The fact that such mechanisms aren't being used to secure devices hints at a deeper problem. Many vendors of

IoT-capable devices don't know how to address these security issues – in fact, it seems likely that many of them have no idea such problems exist. And the root cause of this is that they are simply sourcing the Internet capabilities of their products from third parties. This could be anything from using a library to provide communication facilities, to complete hardware and software stacks that offer off-the-shelf capability that a vendor can simply bolt onto a product. And again, the race to market has a lot to answer for.

“Vendors are looking for minimum viable product – a term that I absolutely do not like one bit,” says Mackey. “And as a result of that, they're looking for the first, least expensive library or solution service that will satisfy the requirements. More often than not, that involves a search on the Internet that comes up with a list of hub projects which will probably satisfy the requirements. They pick one, they incorporate it into their solution, it works and they ship it – without necessarily going through the same security reviews that a larger entity with a history in the software space – like say an Oracle or a Microsoft – might put their software solutions through.”

There is no real relationship between the vendor and the supplier of the component, nor is there any real incentive to audit the component and ensure it isn't full of exploitable holes or back doors. It's purely an effort to reduce work and cost.

There's often little attempt to reduce the attack surface. Mackey gives the example of real-time, firmware-based solutions in Scada systems – that were honed for the task and didn't carry unnecessary baggage – being replaced with more general Linux systems – perhaps on development platforms such as the Raspberry Pi – that offer the cost benefits of a commodity product but may be running unnecessary services that could be vulnerable to attack. When the time comes to deploy, rather than stripping the solution down to its essentials, it's easier to just roll out the full software from the development system and not worry about what vulnerable elements might be going along for the ride.

Too feeble to secure






When it comes to powering the IoT, most people think in terms of small, embedded platforms. These include computers running a full operating system – typically Linux or something similar – with all the capabilities that implies. But just as often it will be something stripped down, perhaps just a microcontroller with a TCP/IP stack added. So doesn't this lack of power pose problems when it comes to layering on security features?

"It really does," says Mackey. "If you take a larger environment, say something based on Intel's Edison or some of the larger ARM processors, they have sufficient memory and compute horsepower to implement, for example, a proper TLS stack. As you move down into the microcontroller realm, they simply don't have the horsepower or RAM available to them to provide proper security, so you need to look at other ways of securing the communication. At that point, you start to see shortcuts being taken where, for example, rather than have a generic TLS stack that's going to include certification revocation and things of that nature, they embed the signatures of the certificates directly into the firmware."

"We've now reached the level where the Microchip PICs of the world can communicate with, say, a sensor or some other interface element, but have a wifi-enabled or network-enabled stack and firmware that can be remotely updated quite successfully"

This isn't to say that microcontrollers can't be secured. Mackey says there are many interesting developments in this area, such as the use of elliptic curve encryption that enables fast encryption on low-power devices. This is particularly suitable in applications where an IoT device might be talking to a service hosted in the cloud, where the decryption side of the process can be enabled with the full power of, say, Google- or Amazon-hosted servers.

Nevertheless, hard-coding resources such as certificates and even, alas, pass-

	 IoT Endpoints	 IoT Infrastructure	 Internet Infrastructure	 Cloud & Datacentres	 Client Devices
Applications	Most Likely Open Source	Most Likely Open Source	Most Likely Open Source	Most Likely Open Source	Most Likely Open Source
Frameworks	Most Likely Open Source	Probably Open Source	Probably Open Source	Probably Open Source	Probably Open Source
Enabling M/W	Most Likely Open Source	Probably Open Source	Probably Open Source	Probably Open Source	Probably Open Source
OS	Most Likely Open Source	Probably Open Source	Probably Open Source	Probably Open Source	Probably Open Source
Firmware	Most Likely Open Source	Probably Open Source	Probably Open Source	Possibly Open Source	Possibly Open Source
Dev Tools	Most Likely Open Source	Probably Open Source	Probably Open Source	Probably Open Source	Probably Open Source
Hardware	Most Likely Open Source	Possibly Open Source	Possibly Open Source	Possibly Open Source	Possibly Open Source

Most Likely Open Source Probably Open Source Possibly Open Source Probably Proprietary

Places where open source software is most likely to be employed in Internet of Things solutions.

words continues to be an all-too-common approach used by vendors looking to at least pay lip service to security. And there are many problems with this approach. For one thing, certificates expire but how do you deal with that if the certificate is hard-coded into the firmware?

"There's a certain threshold at which something like an over-the-air update becomes a viable solution," says Mackey. "The interesting thing is that we've now got to the point where the system on chip (SoC) is sufficiently inexpensive, in terms of its feature/function set, that you have SoCs that are fully functioning, wifi-enabled, compute-enabled devices from a variety of vendors that are sub-\$2-3 per chip. They have a fully functioning network stack, they may have a fully functioning Bluetooth stack, and they have enough compute power and memory that you are really looking at a computing environment that rivals that of a desktop of 12 or 15 years ago. We've now reached the level where the Microchip PICs of the world can communicate with a sensor or some other interface element, but have a wifi-enabled or network-enabled stack and firmware that can be remotely updated quite successfully."

Biggest problems

The Mirai botnet was one of the clearest demonstrations of how weaknesses in IoT devices can be exploited. But many of the threats are less public. Rather than bringing down popular websites and major parts of the Internet's infrastruc-

ture they instead compromise privacy – a case in point being the furore that surrounded the My Friend Cayla doll that was classed by Germany's Federal Network Agency as "illegal espionage apparatus".⁴ So where does Mackey believe the most serious problems lie?

"My primary concern, from a technology perspective, is a lack of transparency with the end consumer of these devices as to what types of data controls and ownership responsibilities need to be in place for them to be truly successful in a secured environment, such that they don't accidentally leak information out there," he says. He gives the example of a 'nannycam', commonly used by parents to keep an eye on childcare professionals. The parents should be the only people able to see the output of the device, he says, and you certainly don't want the video making its way on to the public Internet. The raises the questions of what reasonable (and hopefully effective) steps the vendor has taken to ensure that that's the case.

"That is part of a security review that should happen prior to the devices ever shipping," says Mackey. "It should be part of what a vendor is putting in place in terms of a confidence level that they want to establish with the consumer. It's not just simply shipping a device and collecting some revenue from it and then moving on to the next individual. That's where I see a lot of the real problems."

If vendors regard consumers as people with whom they want to have a valued relationship, or pay more attention to

their corporate reputation, that would go some way to addressing the issues, he feels. “That provides a level of incentive to do the right things with respect to security reviews,” he says, “understanding dependencies and prioritising things like scanning of code for vulnerabilities – things of that nature.”

This seems unlikely in the age of cheap, commodity products, many of which have short shelf lives. And while some form of certification might appear one way to go, Mackey is unconvinced, pointing out that Microsoft tried this with WHQL-signed drivers, yet there are still compliant drivers that are insecure.⁵ Nevertheless, we could do with some properly defined standards.

“As far as there being some accountability standards, I think that’s a worthy endeavour,” Mackey says. “Recently there was a discussion in the US around whether we need to have federal regulations around IoT devices. One of the things they stumbled over was the definition of an IoT device and what type of regulation they might put in place. If we looked at it through a lens similar to what you have with the EU General Data Protection Regulation where there’s a set of responsibilities, a set of ownership requirements, where we want to ensure that data leakage and data privacy don’t just apply at the corporate level but apply to all network-connected devices, that might be an interesting discussion to have.”

Inappropriate language

Inevitably, many of the security issues around IoT come back to what the developers are doing and whether they are going about their tasks in the most appropriate manner. To start with, are they even using the right language?

“Some languages and some platforms are a little more secure than others,” says Mackey. “People use languages that really weren’t designed for an IoT platform.”

He gives the example of NodeJS, which is very popular with hobbyists, being deployed on IoT solutions just because it supports JavaScript. “There’s a reason that compiled languages like C are the

bread-and-butter of anyone making firmware,” he adds, “so there has to be some inherent advantage to that.”

Ignoring those inherent benefits and employing JavaScript purely because it’s easy and available definitely falls into the category of ‘just because you can, doesn’t mean you should’, says Mackey. And that probably extends to the use – or at least, the indiscriminate use – of open source software.

“The realities of open source development are very, very different from the realities of commercial software development,” says Mackey. “One of the biggest challenges that open source creates for IoT devices is the belief that it’s just free software. Somebody implemented this feature or function, I can bring it into my device and it’s magically delicious. The reality is, there’s a level of responsibility.”

“In commercial software, there would be a push mechanism from the vendor to say, ‘here’s a new update to my library, you genuinely need to take that’. With open source, it’s a pull mechanism”

Let’s say that a defect is found in an open source library and that the developer or team behind that library addresses that problem in the right way – working responsibly with whoever found the flaw, fixing the problem and pushing out patches. Does that mean the problem goes away? In most cases, Mackey believes, the answer is no.

“How would the vendor, who had just blindly consumed that componentry, know that the patch exists?” he says. “In commercial software, there would be a push mechanism from the vendor to say, ‘here’s a new update to my library, you genuinely need to take that’. We see that with Microsoft, we see that with Apple and so forth. With open source, it’s a pull mechanism. If you’re not engaged with that community, there’s no way to know that there are any updates, let alone something as serious as a security update.”

The open source community itself needs security policies that define how to interact with the researchers who are

uncovering vulnerabilities, says Mackey. “And we need to have the vendors recognise that they are, in fact, security researchers as well.”

He believes there is an issue with the CVE process as it stands. There’s only a relative handful of CVE Naming Authorities (CNAs) that are authorised to create CVE entries. That can lead to delays or even to vulnerabilities being overlooked. This led to the creation, a couple of years ago, of iwantacve.org, which provides a means of filing information about security issues in open source code.⁶

“That mechanism is something that vendors need to recognise – that they’re part of the solution,” says Mackey. “They can’t have a CVE disclosed against their firmware, or have a security researcher find an issue with the firmware, and have no mechanism to communicate to the broader world that there’s a CVE and people should be updating.”

A genuine problem

While we’re on the subject of researchers, it’s also worth asking how many of the vulnerabilities that are found represent genuine threats. Researchers love to find flaws – apart from anything else it boosts their reputations. And the nature of open source means that it’s easier to find weaknesses by trawling through the source code than it is with proprietary software. But in the real world, how many of these vulnerabilities are being turned into exploits that should really concern us?

“That’s hard to quantify,” says Mackey. “You see things like the BlackHat presentation from the fellows who managed to get into car washes.⁷ You see the attack on the Jeep utility vehicle a couple of years ago.⁸ But one of the things that we have to separate out is, when is there a quasi-closed network. If you take General Motors’ OnStar services, that was an Internet-connected vehicle management system, but it was largely on a closed network. You take the Jeep issue – yes, it was on the Sprint telecommunications network, but it was still fundamentally a closed network. But when we start to get into

broader Internet-connected devices, as far as that closed network is concerned, well, we really can't lean on it."

Honeymoon period

In a sense, the novelty of IoT is one of the things that has prevented more attacks taking place than we've witnessed so far.

"I see organisations that are truly invested in security doing the right things with respect to their devices. My worry is about the smaller vendors, where software security is not a core competency"

"The devices themselves – Mirai and Mirai-like scenarios aside – are enjoying a level of obscurity that every computing paradigm has had," says Mackey. "I remember the days when Linux was great for no other reason than no-one was writing viruses for it, and eventually that changed. I once hosted web properties with an entity that boasted, 'we use Macs as our servers because no-one's written viruses for a Mac'. Eventually that changed. Today, people haven't yet gotten to the point – again, Mirai aside – of really looking at what the potential for an IoT device attack is. We're seeing the beginnings, where the attitude is: 'Well if I can't really do anything useful with this, let's turn it into a botnet and bring down the likes of Dyn'. Those are simple things, but they should be early warnings that it's not safe to just rely on this being the best they can do. This is where they have started. They're clever and they are probably a lot more resourceful than we think they are, and eventually they'll do more harm."

If we're in a kind of honeymoon period, how does Mackey see this playing out over the next couple of years?

"I see organisations that are truly invested in security – Google, for example – doing the right things with respect to their devices," he says. "I fully expect that Amazon and other large vendors are doing exactly the same. My worry is about the smaller vendors, where software security is not a core competency – for example, organisations that are in the business of making mechanical locks deciding they want to have wifi-enabled locks, or Bluetooth-enabled locks, so they spin up a small group of engineers to produce that lock. Then it ships and they don't necessarily have the competencies to do the security reviews. They fall victim to some vulnerability; they have to resolve that but don't really have a mechanism for doing so. And then couple that with the longevity of the devices. The house that I bought two years ago had the original locks from 25 years prior. We're not accustomed to thinking of computing devices having that type of lifespan."

So is the situation going to get a lot worse before it gets better?

"I think that's a very strong probability," says Mackey. "What we may see is something equivalent to how the spam problem evolved. There was a period where people had their email clients just effectively naked on the Internet. Then ISPs recognised that home users had no rational reason to be running SMTP servers, so they started blocking port 25, and there started to be layers of reality. I suspect that we'll have to go through that phase to deal with the legacy devices out there before things ultimately get better. We'll end up in a similar protocol/port-type arms race for the next two, three years while we deal with these legacy devices while also doing the right thing with respect to security and general Internet hygiene."

About the author

Steve Mansfield-Devine is a freelance journalist specialising in information security. He is the editor of Network Security and its sister publication Computer Fraud & Security.

References

1. 'Ladder logic'. Wikipedia. Accessed Feb 2018. https://en.wikipedia.org/wiki/Ladder_logic.
2. 'Mirai (malware)'. Wikipedia. Accessed Feb 2018. [https://en.wikipedia.org/wiki/Mirai_\(malware\)](https://en.wikipedia.org/wiki/Mirai_(malware)).
3. 'Common Vulnerabilities and Exposures'. Mitre. Accessed Feb 2018. <https://cve.mitre.org/>.
4. Oltermann, Philip. 'German parents told to destroy doll that can spy on children'. The Guardian, 17 Feb 2017. Accessed Feb 2018. <https://www.theguardian.com/world/2017/feb/17/german-parents-told-to-destroy-my-friend-cayla-doll-spy-on-children>.
5. 'WHQL Release Signature'. Microsoft, 20 Apr 2017. Accessed Feb 2018. <https://docs.microsoft.com/en-us/windows-hardware/drivers/install/whql-release-signature>.
6. 'Distributed Weakness Filing (DWF) CVE Request form for PUBLIC issues in OpenSource software v5.0'. Via Google Docs. Accessed Feb 2018. iwantacve.org.
7. Thomson, Iain. 'Hackers can turn web-connected car washes into horrible death traps'. The Register, 27 Jul 2017. Accessed Feb 2018. https://www.theregister.co.uk/2017/07/27/killer_car_wash/.
8. Greenberg, Andy. 'Hackers remotely kill a Jeep on the highway – with me in it'. Wired, 21 Jul 2015. Accessed Feb 2018. <https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>.