John Leach Information Security Ltd

An Engineering Approach to Security Design

An article for publication
Written by
Dr John Leach
John Leach Information Security Ltd

24th July 2003

Dr John Leach
John Leach Information Security Limited
☎ = (+44) (0)1264 332 477
✆ = (+44) (0)7734 311 567
📠 = john.leach@jlis.co.uk

AN ENGINEERING APPROACH TO SECURITY DESIGN

IT Security has been practised as a dark art for too long. We should treat it as an engineering discipline and reset our expectations about how security systems should be designed and evaluated. All it would take is a fresh approach, the right metrics and a little competent analysis. This is how it might work.

There is no reason why security systems should not be designed by security engineers in much the same way that, say, bridges are designed by structural engineers.

To design a bridge, a structural engineer works with a number of quantifiable factors. He will know what loads the bridge will need to carry and what stability and budget criteria it will need to satisfy. He will bring in data describing the bridge's operational environment such as the wind speeds and air temperatures to which the bridge will be exposed given its planned location. And he will draw on detailed results showing the strength and structural characteristics of different types of steel. Equipped with this data, he will proceed to calculate what type of steel he needs and how thick the steel needs to be at each point within his design.

A security engineer should be able to design security systems in a similar way. He would be told by the system owners the operational targets for the system being protected, and the security targets, in the form of security SLAs, that the business would like the system to achieve. He would agree with them the metrics by which he would test, and they would assess, whether or not the end system achieved their security goals. The security engineer would know, from standard analysis, what threats the security system would be buffeted by given the environment, Internet or corporate intranet say, to which it will be exposed. And from security research sources he would get a quantified assessment of the typical level and intensity of those threats prevailing in that environment today. Then he would proceed to calculate by simple analysis what type of security steel he needed to build his security system from, and how much of each, given detailed results showing the strength and resilience of different types of security steel.

He could go further. The security engineer could work out the range of operational environments within which the protected system could comfortably

be run, and how much additional security steel it would need if the system were to be adapted for a more hostile threat environment or for processing more highly classified data. He could work out what monitoring was needed so that system operators could detect in good time if the operational system started to show signs of security strain, and he would provide technical input into the safety plans so that incidents didn't turn into disasters.

Yet who today can design security systems that way? Imagine a CIO approaching his Chief Security Officer and saying that their new CEO wanted the company to be more risk averse this year than last. How nice if the CSO could tell them both exactly how much thicker their anti-virus and vulnerability patching security steel would need to be to achieve the CEO's improvement target? He could tell them what level of risk the company was currently running of being breached by a virus or holed by a worm, and what additional security steel would be needed to improve that to, say, no worse than 1% chance per month. It is a straightforward scenario, and company top management should be able to expect their security chiefs to give them this type of constructive response.

So, what stops security designers from being security engineers in this way? The lack of the right type of data is part of the answer. But that just begs the next question: why don't we have that data? It isn't the lack of database technology or data mining techniques that stops us. We should easily enough be able to gather ample data to tell us accurately whether our staff cause more security incidents in our West Coast offices or our New England offices, or whether our staff in our European offices update their desktop antivirus signatures more frequently than those in our SE Asia office.

The data is there to be gathered but it seems we are not in the practice of gathering it. We haven't started to gather it because we haven't yet articulated clearly what questions we want the gathered data to answer. For this reason, we are still taking guesses at what data it might be useful to gather and don't have a clear idea what analyses we might wish to apply. If we can't quantify clearly and simply what security SLAs we want our security solutions to satisfy, then the best we can expect is to continue as in the past: security managers exhorting people to build in more security and trusting that doing more will translate somehow into them having fewer security headaches to endure.

For IT Security to become an engineering discipline, we need to start with a clear understanding of exactly what it is we want our security solutions to

achieve and how we plan to evaluate our level of success. That will tell us what data we need to gather and how to analyse it.

So, what is it we are trying to achieve with our security solutions? A CSO responsible for the protection of an IT infrastructure supporting a variety of mission critical and routine business systems is not measured by the board on how many penetration tests get run in a year or how many pages of security standards get updated each month. He is measured by the number and intensity of the security headaches the IT service sees and the level of disruption the business experiences.

In this case, maybe his security goals should include such things as:

Security Goal	Interpretation
Be able to withstand the "once in a year" malware event without serious damage to the IT infrastructure and with disruptions to the IT service kept to no more than one hour at peak onslaught.	This might mean being able to withstand the next Code Red, Nimda, or SQL Slammer storm with no worse a disruption to the IT service than having to close an Internet gateway for at most one hour in the early stages while the nature of the storm is assessed and checks made that patch levels have been maintained.
Be able to deal with 5,000 users per day and a once-per-quarter peak in user security failures (users getting locked out of their accounts, users visiting unauthorised web sites, users trying to gain unauthorised access to sensitive servers) without any perceived disruption to the IT service.	This might mean being able to get legitimate users back into their accounts within 10 minutes of them locking themselves out, and assessing and prioritising apparent access violations within 20 minutes of them occurring.
Be able to detect any signs of unusual activity or heightened security strain within the IT infrastructure with enough notice that, at least 95% of the time, the problem can be diagnosed and appropriate steps taken before there is any perceived impact on the IT service.	This would mean knowing what sort of security strains to look out for, the kinds of events or attacks expected to be causing them, and how much time was available before the IT service would start to be impacted. This will tell security operators how promptly they had to detect growing signs of security strain, and how much time they would have for responding to a major alert.

If these are the type of security SLAs security solutions might be engineered to achieve, how might we test whether the security engineer had achieved them? As always, the real test will be how well Operations responds when "Son of Nimda" comes along or when the top systems administrator is found to have created ten unauthorised NT domains for his own personal (and highly illegal) use on the corporate network. But we should want to have a good idea of how well protected the IT infrastructure is and whether these security SLAs have been met before that unforgiving live test comes along.

The engineering questions these security goals suggest and, hence, the answers that we might want to gather data to resolve are, for example:

Question	Answer
How soon after a software vulnerability is discovered does a piece of exploitative malware turn up?	Six months ago the answer would have been that Code Red was the quickest seen, appearing 31 days after the vulnerability was discovered and 28 days after the patch was released. We would like to know, is that still the right answer? Is there reason to believe that malware writers are getting faster off the mark? These answers will tell the CSO how promptly security patches need to be applied to the IT infrastructure.
How quickly does the probability of exposure to a new virulent virus rise as a function of days since the virus was first seen in the wild?	For BugBear, a home PC user receiving on average 20 e-mails per day would have faced a 20% probability of exposure to BugBear after 5 days and a 50% probability of exposure after 11 days. A CSO will need to calculate his company's exposure profile given the number of incoming e-mails per day the corporate e-mail service handles. That will tell the CSO how rapidly that mail server's anti-virus signature files needed to be updated if the chance of getting infected by each new strongly active virus is to be kept below, say, one in 50.
What proportion of user accesses to the company's most sensitive systems are actually unauthorised accesses where someone has used someone else's user-id?	Unless the user community is extremely vigilant, then probably the only way to find this out is by doing checks of the system access logs showing user-id usage against the building access logs showing the user-id owner's daily arrival and departure times, and the user's holiday records. For most organisations, a check like that would have to be done manually. This means doing no more than a limited number of spot checks. In that case, the CSO needs to know how many spot checks to do each month. Simply letting staff know that spot checks are performed and the results followed up will lead to a fall in the level of unauthorised access. The residual level of unauthorised access will help the CSO to judge which situations warrant tokens being used for user authentication rather than just passwords.

These questions would, in principle, be simple to answer with a modest amount of analysis if our security engineers had access to the right data. But I would hazard a guess that there are very few organisations today that could lay their hands on this type of data, or equivalent data for other similarly useful questions. If the data is external data, such as virus and vulnerability data, considerable hunting is required to find it. If the data is internal data about one's own user community and their security behaviours, most organisations could not lay their hands on such data because they have never seen a reason to collect it. Yet, as an industry, we have been saying for a while that poor security behaviour rather than poor security solutions are behind many of the security problems we experience. Programmes to improve people's security attitudes and security behaviours must be worthwhile supporting, so what stops people gathering the data to substantiate that case?

We would be able to make good business cases for our favoured security initiatives if we approached IT security as an engineering discipline. There are a number of steps that the IT Security industry needs to take before it can hope to achieve that, and getting the right type of data together is one part of that.

The first step is to take a fresh approach. We have relied on familiar principles and established practices for so long that we can't be sure they are still valid in today's environment. We need to put our faith in data and the evidence that data can provide. We need to let the data tell us which are the security problems we need to attend to, and we need to learn how to analyse data reliably rather than superficially. I recently saw someone interpret some virus exposure data and come to the conclusion that the probability of a user having been exposed to a particular virus after nine days was 200%. Common sense suggests we should be able to do better than that.

The second step is to develop a framework for describing the security characteristics of threats and solutions in terms that can be quantified. Do we use Confidentiality / Integrity / Availability as our three dimensions for describing security threats? And if we do, what are our yardsticks? Availability might be measured in terms of overall up time over the month, but what would be our yardstick for Confidentiality? I think C, I, A are three of our security dimensions, but they are just the first three out of a set of maybe eight or nine. We need an agreed definition of what the security dimensions are and what their yardsticks should be so that our security terms can be quantified. That would give us a standard set of terms for expressing our security SLAs, measuring the

constituents and intensity of a threat environment, and gauging the strengths of our security steel, the security components we build security solution from.

The third step is to develop mechanisms for quantifying risk aversion and how much insecurity business management is prepared to tolerate for a given system or environment. If we had yardsticks for the security dimensions, tools could be developed to translate a business management's view of risk into insecurity tolerances, how much insecurity they would be prepared to tolerate for the business. These would give us the Security SLAs that the system then needed to satisfy.

The fourth step is to calibrate security steel, our security components. Once an engineer knows how much resistance needs to be put up against the prevailing threat environment, he will want to calculate how much security steel that would take. Can he get enough threat resistance from a light-touch security solution, or does he need to build something more substantial to meet the security targets given the pressure of those threats? How much resistance to viruses would he get from a well maintained mail server anti-virus product, and would he need to spend money on an additional layer of desktop anti-virus to achieve his virus infection SLA or would that not be necessary?

Are these four steps achievable? I think they are. I think we just need to be prepared to cast off our old security attitudes and start afresh. And is now the time to do this? I definitely believe it is. There is a huge call right now for measuring security Return on Investment (RoI), and I doubt the demand will abate even when the economy recovers and people are not so tightly squeezed for funding or resources. The demand for a presentable Security RoI is just another way for the business to ask us to stop pretending that IT Security is a special type of magic practiced by a chosen few and to start asking security designers to behave like the security engineers they should be.

Prepared By:
Dr John Leach
John Leach Information Security Limited
☎ = (+44) (0)1264 332 477
✆ = (+44) (0)7734 311 567
✉ = john.leach@jlis.co.uk