SPRING 2009

BCS

www.bcs.org/security

INFORMATION SECURITY NOW

Securing software

Stop programs concealing their real identity

SECURE DEVELOPMENT

10 best practices to shore up your code

SHUT THAT DOOR

How to spot and then close any back doors in your software

SOFTWARE ASSURANCE

Making sure that software is safe from the start

SUPPORTED BY THE BCS INFORMATION SECURITY SPECIALIST GROUP

Information Security MSc

Flexible learning for everyone

We have extended the way in which Royal Holloway's internationally recognised MSc is offered.

- **CPD/CPE Modules:** Most MSc modules are now available as stand-alone courses of one week's duration (Block Mode). These modules may be taken with or without an examination.

As a result the MSc now has the following traditional delivery modes:

Full-time, one year, on campus; **Part-time**, two years, on campus; **Block Mode**, two years, on or off campus; **Distance Learning**, up to four years via the Virtual Learning Environment.

The introduction of CPD modules has enabled us to introduce even more flexibility into our methods of delivery.

- **Latest innovation** – 'Mix and Match' degree programmes. It is now possible to obtain the MSc by accumulating modules by any delivery method listed above (maximum period seven years).
- **Postgraduate Diploma** – each module is also available in condensed mode and taught as a one, two or three-day training course offered by QCC Training Ltd. Students may follow a structured programme of these courses and then undertake an MSc level project to obtain the Postgraduate Diploma in Information Security.

Royal Holloway
University of London

Information Security Group
www.isg.rhul.ac.uk
p.stoner@rhul.ac.uk
z.ciechanowicz@rhul.ac.uk
T: 01784 443101

ISNOW is the quarterly magazine of the BCS Security Forum, incorporating the Information Security Specialist Group. It can also be viewed online at:
www.bcs.org/security/isnow

EDITORIAL TEAM
Henry Tucker Editor
Brian Runciman Managing Editor

DESIGN TEAM
Marc Arbuckle Art Editor
David Williams Graphic Assistant

Registered Charity No 292786

Printed in Great Britain by:
Inter Print, Swindon, Wiltshire.
ISSN 1752-2455. Volume 3, number 3.

The British Computer Society
First Floor, Block D, North Star House,
North Star Avenue, Swindon SN2 1FA, UK
tel +44 (0)1793 417 417;
fax +44 (0)1793 417 444;
www.bcs.org
Incorporated by Royal Charter 1984.

ISNOW | ISSG PERSPECTIVE

Gareth Niblett, chairman of the ISSG, says that now is the time to make software secure.

Secure, stable and reliable software is a rare commodity, and one that most can't actually buy. Although we may feel that those who profit from selling us software not up to the task should be held liable and sued, doing so could also expose open source developers to unacceptable risks when giving us their software.

Unreliable and insecure software is due to a variety of factors, from the lack of academic focus within software engineering and computer science courses, to the development approach that our IT professionals are expected to adopt when programming in-house systems or commercial applications.

Even the formal accreditation of systems, e.g. common criteria, may not detect or prevent software vulnerabilities from arising; it can even compound the issue by forcing a decision to be made between operating a vulnerable but accredited system and an upgraded but unaccredited system.

SANS and CWE have listed the top 25 most dangerous programming errors (**www.sans.org/top25errors**), which cover the actual programming errors made by developers that lead to the vulnerabilities that software may be susceptible to, and provides useful and authoritative information on mitigation.

Security software even provides ready examples of how to not do it and the formal methods of safety critical systems may be overkill for most commercial offerings. Something needs to be done to improve the security, stability and reliability of software where more features are delivered ever more rapidly.

We need a 'secure by design' approach, where we seek to minimise the existence and impact of vulnerabilities and other bugs. Secure applications can only come from a top-down design and development ethos, integrated with a robust software development life cycle (SDLC) that includes structured testing.

ISNOW | CONTENTS

FURTHER INFORMATION

Information Security Specialist Group: www.bcs-issg.org.uk
Information Risk Management and Assurance Specialist Group: www.bcs.org/groups/irma **BCS Security Portal:** www.bcs.org/security
***ISNOW* online:** www.bcs.org/forum/isnow

Shut that door

Malicious backdoors are designed by attackers to avoid detection by traditional security tools such as firewalls, intrusion detection systems (IDS), and anti-virus software. To these security tools, backdoor traffic looks identical to typical application traffic. Mike Puglia from Veracode explains why you need to shut the backdoor.

Unintentional backdoors are either introduced due to programming errors or an unawareness of the threat that backdoors present. For example, programmers routinely and purposefully insert a backdoor into an application, in order to access that application later for remote troubleshooting.

As the complexity of modern software applications increases, along with the use of components assembled from reusable binary components, backdoors can easily circumvent even the best of QA cycles. The need for an accurate and complete approach to software security testing is more pressing than ever.

At the same time, applications have become a target-rich environment for those seeking monetary gain through the misuse of personal information. These trends have increased the occurrence of backdoors being inserted into applications.

Binary-level application testing or compiled-code analysis makes it possible to examine the entire application in its final delivered form. It enables the most complete detection of backdoors that most often can only be found in application binaries.

An application backdoor is a segment of code that provides an alternative, non-standard entry into an application and can permit functionality not intended by the application's designers. Application backdoors can be inserted unintentionally, intentionally for good reasons, or intentionally for malicious reasons. A common characteristic of backdoors is that they are often planted into an application with the intent that they will be exploited at a later point in time.

While backdoors are becoming more common, their origins are varied. Backdoors are not always introduced with malicious intent. For example, conscientious programmers may build a backdoor into an application for the purpose of remote troubleshooting, an activity that is more common in today's dispersed, interconnected application environments. Less-careful programmers can unintentionally introduce backdoors by re-using code that has not been checked, or simply by being unaware of the backdoor exploitability factor in certain types of code.

Undetected backdoors, regardless of their origin or intent, can give attackers access to data, including customer and company data, and allow them to steal or manipulate that data. This obviously exposes the application developers and their companies to potentially serious risks.

In the past, manual code reviews or automated reviews of source code were used in an effort to detect backdoors. However, these efforts have resulted in limited success due to both the high cost of manual reviews and the high false negative rate of source code analysis tools. Source code analysis tools miss significant portions of the software such as third-party libraries, API calls and code injected via the tool chain during compile time, making the technique unreliable at best.

Recently, new technology has enabled organisations to overcome the limitations of source code testing, by using static binary analysis to provide a complete way to cover application security. Static binary analysis of applications represents a superset of source code analysis.

Three types of backdoors

System backdoors allow access to data and processes at the operating system root

level. They are typically created by an attacker who has compromised a system in order to retain access to the system later on. Examples of system backdoors include rootkits, remote access software, malware such as Trojans or bots and deliberate system misconfiguration.

Crypto backdoors are intentionally designed weaknesses in a cryptosubsystem for particular keys or messages that will allow the attacker to gain access to clear-text messages. Application backdoors are sections of legitimate software modified to bypass security mechanisms under certain conditions.

Application backdoors are sections of legitimate software modified to bypass security mechanisms under certain conditions. These legitimate programs are meant to be installed and running on a system with the full knowledge and approval of the system operator.

Application backdoors can result in the compromise of the data and transactions performed by an application and may also result in a compromise of the complete system.

Special credentials

In these cases, the attacker inserts logic and special access credentials into the program code, typically in the form of a special user name, password, and password hash or key. One company had such vulnerability in its popular commercial database program; it went undetected for seven years. This vulnerability could have been successfully detected using a combination of static detection strategies, including analysis of variables that look like usernames or passwords, identifying static variables that look like hashes, and identifying static variables that look like cryptographic keys.

Hidden functionality

Attackers use these methods to issue commands or authenticate without needing to perform the designed authentication procedure. More advanced attackers even include a special check so that there is some protection from anyone using the backdoor. In 2007, this type of backdoor was inserted in a popular blog publishing program, WordPress, which led to a recall of an entire version of the software. Techniques for identifying this vulnerability should include inspecting the application for command injection vulnerabilities, recognising patterns in scripting languages, comparing server-side code with client-side code, identifying potential operating system command injection vectors, and identifying static variables that look like application commands.

Unintended network activity

The attacker can exploit networks by listening on undocumented ports, making outbound connections to establish a command/control channel, or leaking sensitive information over the network. In many cases, these backdoors are strengthened with rootkits, which are of further concern.

In 2002, a backdoor of this type was inserted in the source code distribution of a popular network sniffer. It established a connection to an IP address and listened for commands.

It is possible to detect this type of activity using dynamic network utility testing, but only if the specific run conditions, under which the program will activate, are duplicated. A number of static analysis techniques can be used to detect this code, including identifying inbound and outbound connections, analysing network activity that references a hard-coded IP address or port, identifying potential information leaks, analysing data flows, examining import tables, and more. The goal is to identify anomalies, then analyse the binary code in more depth.

Parameters that assign certain privileges to a process, influence task scheduling, or restrict operations on memory pages represent a higher security risk. In 2003, an attacker attempted to insert a backdoor into the Linux kernel. Fortunately, the backdoor was noticed and removed before users were affected. The method the attacker used wasn't successful in this case, but it revealed a very subtle technique involving just a couple of lines of code that were activated when a particular process was initiated. The technique used by this attacker could be mistaken for a common, usually benign, programming flaw - using assign instead of compare. Static analysis could be used to identify all instances of this particular behaviour, taking into account the instances where this string of code is intentional.

Automated source code analysis.

Several tools are now commercially available to review large and complex applications without manually going through the code line-by-line. These tools are good at finding secure coding flaws, such as memory usage errors, input validation errors, or mishandled error conditions that an attacker can use to compromise the software.

However, they require access to source code, which many not always be feasible and doesn't take into consideration backdoors that may be injected during compile time, via APIs or hidden in third party libraries. Additionally, this method does not detect most types of application backdoors, precisely because they are designed to be hidden in the source code and only become exploitable after compile time. Static binary, byte-code or compiled analysis examines a compiled form of an application or component to create a complete picture of real-world vulnerabilities. It is important to understand how its two main aspects: binary and static separate it from alternative methods.

Commercial off-the-shelf (COTS) programs can be a problem. An application security expert recently analyzed 100 programs. He randomly selected 100 COTS and open source application packages, analyzing those using reverse-engineering tools. Upon tabulating the results, he found that:

- 79 packages had dead code;
- 23 packages had unwanted code;
- 89 packages had suspicious behaviours;
- 76 packages had possible maliciou code.

Of these:

- 21 packages had known worms, Trojans, rootkits, etc.
- 69 packages had possible worms, Trojans, rootkits, etc.

Whether intentional or not, security vulnerabilities present in an application can leave an enterprise or government organisation open to serious legal or commercial risk. Given the complexity, age, and varied pedigree of code that characterises the typical application, source code analysis is incomplete. Once backdoors and malicious code are inserted, they are impossible to detect without binary application testing analysis. On-demand, static binary application testing for backdoors gives developers and enterprises a complete solution to testing their applications for critical security vulnerabilities.

To encrypt, or not to encrypt: that is the question...

Charlotte Walker-Osborn, technology partner, and Aonghus Martin, technology solicitor, from Eversheds LLP, discuss the English legal position in relation to encryption.

More than 8,500 laptops are left in UK airports and over 10,000 are left in London taxis every year. Human error and the increasing adoption of portable technology inevitably means these figures are unlikely to decrease. The Information Commissioner's Office (ICO) is starting to get tough, giving clear messages that data controllers (in this context employers) must encrypt certain personal data.

It had previously been a question for consideration as to whether, under the 7th Principle of the Data Protection Act 1998, which requires data controllers to take 'appropriate technical and organisational measures against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data' meant that encryption technology should be used. There was little guidance expressly addressing this point. However, there have been a number of high profile companies who have received enforcement notices which specifically detail a failure to encrypt as being a breach of the DPA and DP principles.

Data protection law is therefore, largely by way of enforcement notices and guidance (issued on security and encryption both in 2007 and 2008), beginning to impose certain specific obligations and recommendations on companies, in relation to encryption.

Although these are not definitively stated, it seems pretty clear that where personal data (data which can identify a living individual which can include name, HR records etc.) is placed on mobile devices, encryption must now be used. Otherwise, there is currently no strict legal obligation to encrypt personal data, although it may be helpful in some cases to do so voluntarily, for example where such data has to be emailed to a higher risk country or is particularly sensitive or potentially valuable or damaging. There is no specific legislation imposing obligations to encrypt confidential information not containing personal data in the UK.

However, you should consider using encryption in relation to relevant technology to protect important business information. This is likely to include certain HR information (which also gets squarely caught by the DPA), plus board information, certain financial information such as pricing, confidential information on customers and other important confidential information.

So, outside of mobile devices, you should be making judgments around the types of data that should be encrypted as well as keeping an eye on ICO and other security guidelines on this area.

With newly granted powers given to the Information Commissioner now is a good time to undertake a review of the security applied to personal data being processed within your organisation and how this is treated in your contracts, for example with service providers, including your encryption obligations. This will assist in guarding against a security breach and the resultant adverse publicity, reputational damage and loss of customer confidence which flows from such incidents.

In addition, buoyed by a plethora of recent data loss incidents and new powers recently introduced, such as the ability to levy fines for serious breaches, it is clear that the Information Commissioner will be looking to flex his new found muscles.

FREE ENTRY

GC live

9-10 June 2009 | Earls Court, London

Procuring Public Sector ICT

Stay connected at the UK's leading event for Public Sector ICT

GC Live is your best opportunity this year to discover the latest solutions that are driving efficiency and reducing costs across the public sector.

- Connect to over 100 innovative ICT suppliers in just two days, including; BT, Canon UK, Corporate Document Services, Hays IT, ntl Telewest Business, Siemens Plc and many more.
- Connect to the hottest debates and the most important issues affecting the future of public sector ICT in the Public Sector Talks.
- Connect to a wealth of hands on ICT solutions which will immediately benefit your day to day role in the Government Computing Talks.

Register today at **gc-live.com**

Does your procurement colleague know about **Procurement Solutions Live**? It is the UK's official government procurement event. Over 100 approved buying solutions suppliers will be showcasing their products and services. **Visit procurementsolutions.gov.uk**

Part of

The Public Procurement show

9-10 June 2009 | Earls Court, London

Procuring New Solutions

Official media partner: LGC Local Government Chronicle

Supported by: ogc buying.solutions

In association with: theguardian, BCS

Organised by: kable

Securing information systems

The IT industry is at the heart of developing future resilient information systems says Andrew Tyrer from the Technology Strategy Board.

Picture the scene: it's a typical day in your business or private life. You've woken up, checked the television or radio news for transport updates, read emails on your BlackBerry, tweaked your SatNav system on the way to school, work, or the supermarket.

You're barely an hour into your day, and yet, you've become reliant on complex systems underpinned by IT. We're at the stage in modern society where you simply can't go back to paper information. What a wonderful technology-inspired world we live in. But wait, these systems cannot fail, can they?

The answer, of course, is yes they can. As our dependence on information systems increases, so does the risk of these complicated tools failing through capacity overload, human intervention, or natural disaster. In fact, not only do we depend on these systems, but the systems themselves are also heavily reliant on each other. In the home we are running multiple internet connections, home entertainment systems, digital televisions and telephone lines that converge into a single set top box, dealing with more complex information year upon year. As an information system matures, it converges with many other technologies due to the demand for increased agility, virtualisation and interconnection. The end result is an unplanned 'system of systems' where functionality overrides resilience, leading to security concerns. If this fails, it can take out many systems at once.

For example, a significant systems failure was the electrical blackout of the eastern seaboard of the United States in August 2003. This breakdown in continuity lasted for more than 48 hours and affected more than 50 million people. It was suggested that the initial event, which lead to a chain reaction, started at a power plant in Ohio. A breakdown in the computer control system failed to detect a small electrical problem and rectify it. This small scale local event cascaded into a major outage for a large population of eastern United States and Canada.

Closer to home we witnessed the Buncefield oil explosion and subsequent fires in 2005. What was perceived as an environmental peril soon became an IT information problem. The fires caused damage to IT data storage company Northgate Information Systems' equipment. The knock-on effect led to Addenbrooke's Hospital IT-centred patient admission system failing, causing major disruption.

Safeguarding our complex information systems

To counteract these security fears, we need innovative and technical solutions to enable systems to be managed – to mitigate risk. These systems will get even more complex in the future, so there is an element of the unknown. We need expertise today to start predicting future security problems. We need to start taking a prevention approach, not cure.

We need collaborations across diverse industry sectors such as transport, healthcare, engineering and finance - all underpinned by IT expertise. We're facing a massive societal and business challenge, but we believe the UK has the expertise to tackle this challenge – and the Technology Strategy Board is at the forefront of this societal problem.

We are working with the Centre for the Protection of National Infrastructure and the Engineering and Physical Sciences Research Council to allocate £6m in research funding to secure our business information systems.

During this funding period, we want organisations with the necessary skills to develop tools, techniques and services to tackle the ever-increasing threat to our information systems. This investment will directly target the complexity and dependency challenges associated with intricate information systems that UK government and businesses use daily.

This funding competition will address innovative solutions for making our information infrastructure more robust. This could include the development of real-time predictive models with particular emphasis on interdependency analysis and supply chains.

No 'silver bullet' solution

We don't see there being a 'one size fits all' solution, but we welcome innovative ideas that will address high level challenges that include:

- increasing understanding and management of complex interdependent IT infrastructures and systems;
- development of models focusing on real-world practical applications to enable SMEs and large companies to secure their information systems;
- producing systems with better scope for data capture, security and data segregation across industries such as healthcare, assisted living, intelligent transport;
- bringing together diverse groups such as IT professionals, academics, health professionals, economists, transport planners and insurance professionals to share knowledge and ideas;
- making software more secure, and therefore less susceptible to security vulnerabilities and attacks.

We see these challenges being met by pioneering thinkers within the information security and IT community. We understand that staff from SMEs are very busy, often working on their own, which is why the Technology Strategy Board is offering its full support to the SME community to encourage individuals to form collaborations and apply for this funding. SMEs who successfully apply for competition funding will be able to keep and exploit the intellectual property they develop from their work. This will be financially beneficial, especially if an entrant's work is produced for a new burgeoning commercial market.

We want to make it clear that this competition is not about funding research

that won't produce tangible results. The competition offers the only UK public money currently available to address the security of society's complex information systems, so naturally we want to see a return from our investment.

It's vital that research proposals clearly demonstrate positive economic and business impact, coupled with environmental and social sustainability. We strongly encourage projects that can demonstrate tangible benefits across business sectors. It's essential that the research outputs could, for example, benefit the banking industry as well as transport planning and healthcare systems.

The solutions

We are not going to pretend that solving system security weaknesses will be easy, but we are confident we have the expertise to benefit services that make our lives easier to live, in the home, in the work-place and on roads we travel upon.

In the home, we see this research making our internet connections safer. Also, as our population ages, we could see the funding design a safer home where technology can support our wellbeing. The research funding could enable systems to remain robust, avoiding downtime, allowing constant monitoring of a person's health and activities.

In the workplace research has many potentially successful and beneficial applications. The banking industry could benefit from better software that predicts risks from cyber attacks. In healthcare industries, better information systems to ensure patient's records are maintained securely can be designed and implemented. Transport systems could benefit from more robust IT systems.

These are possible solutions to improving our complex information systems. We know there are many others. We're challenging industry to play a major part in making our business and private lives more efficient through secure systems.

We have become a digitally-dependent society; the days of paper systems are well and truly a thing of the past, so we need to collaborate, to strengthen our information-based society, for the now and in the future.

Further information

www.networksecurityip.org

Software assurance with SAMM

Within the security industry, a recent trend is a focus around considering security from the start during the development of applications. This is by no means a new concept; it simply was not a focus in the industry as vulnerabilities within infrastructure (network, operating systems, and databases) were what were being exploited at the time.

However, as these vulnerabilities were remediated and defence mechanisms at those levels improved, attackers shifted their focus to other avenues such as phishing and application level vulnerabilities. In addition, as we are more connected to the internet via high speed connections, more and more organisations have migrated their software applications to the internet for wider accessibility. As a result, focus within the security industry has shifted to software security.

An example of this shift in focus can be seen in the evolution of the payment card industry data security standard (PCI-DSS). The earlier version of this standard (when it was called Visa Cardholder Information Security Program) mentions that organisations should follow change control procedures for their software modifications as best practice items under a few of the requirements. In contrast, the latest version of the standard (v1.2) dedicates the majority of one of its 12 requirement areas (requirement 6) to the security of software applications.

When it comes to creating software it needs to be done with a secure base. Matt Bartoldus from Gotham Digital Science has the assurance.

Software security issues

As an application security consultancy, a question we get asked a lot is what issues are you seeing your clients? What do you help them with? Organisations have differing drivers for security such as compliance (DPA, SOX, PCI-DSS), governance, protection against crime, brand reputation, etc, but the primary issue our clients face is figuring out how to manage and operate information security processes effectively within their organisation. Due to the pervasive nature of information security controls, they must be embedded within business and IT processes.

Information security within software development is a prime example of this.

Many organisations struggle with how to

incorporate information security into their software development projects effectively, largely due to time and budget constraints. Some of the more common questions we get asked include:

- What does 'it' look like?
- How can we understand and manage 'this'?
- Do we have enough resources / skills to do 'this'?
- How does 'this' fit in with the security function, shouldn't they do 'it'?
- We are used to security projects that implement tools or systems but now we need to change our processes?
- Isn't there an established method or model for all 'this'?

You may have noticed that I have referred to information security within the software development process as either 'this' or 'it'. Currently, information security within the software development process is still a discipline in definition. There have been many guides and methodologies published in recent years by both public and private sector organisations. While many of these differ in naming conventions and presentation, they are all very similar in nature. While there are many definitions and acronyms for this discipline, it is commonly referred to as software assurance.

OpenSAMM

Recently, an attempt has been made to capture the elements of the discipline of software assurance within a framework that varying organisations can use as either reference, to assess where they are in terms of software assurance maturity, or to build a roadmap to implement security into their software development process. My company has adopted the software assurance maturity model (SAMM) in order to put all of 'this' into a framework that the business can understand. It has been released under an open license, the creative commons attribution-share alike license, and is set to be an official project within the open web application security project (OWASP). As such, it is also commonly referred to as OpenSAMM or simply SAMM. This body of work was written and edited by a group of software security experts within the industry under the support and funding of Fortify Software, Inc., a leading provider of application lifecycle security and source code analysis tools. It can be downloaded from

www.opensamm.org

Here I will discuss some of the core activities within SAMM but firstly, I would like to mention what SAMM is not. SAMM is not a prescriptive 'how to' document for software assurance; nor is it a 'one size fits all' methodology; nor is it an audit checklist around secure software development.

Core activities of OpenSAMM

As mentioned before, there have been many guides and methodologies published around software assurance over the years with common areas addressed. The following are some of the key core activities within SAMM in which we work with our clients to achieve software assurance:

- Education and guidance - EG. This includes focusing on the right security training for the right personnel involved within all stages of the software development process. SAMM goes as far as suggesting information security related certifications in areas of software development which can be a good motivator for technical staff.
- Security requirements – SR. In order to plan for information security to be built in to software, it has to be detailed as requirements so they can be developed and tested in the same way as other functional or non-functional requirements. SAMM recognises that not all software development projects are the same. Security requirements need to be tailored based on several risk factors such as the type of software being developed, data that will be processed or who will have access.
- Threat modelling – TM. Threat modelling is an activity performed in order to focus on what the threats are to an application and likely attacks it may face once developed and deployed. Information security requirements are then matched up against the identified threats in order to determine whether the security requirements have addressed all identified threats appropriately. This is done early in the software development process usually prior to when functional and technical specifications have been finalised.
- Architecture review – AR. The architecture review activity is focused on the review of software designs and architecture models for potential security related deficiencies. This allows an organisation to detect architecture-level issues early in software development to avoid potentially large costs from refactoring down the road. The security requirements developed for the project as well as either the organisation's security architecture or best practices are used as the basis for the review.
- Standards and compliance – SC. In order to meet the information security requirements of the organisation (both internal and external), software development projects need to have an understanding of such requirements as well as have periodic compliance reviews at the appropriate project gateways.
- Security testing – ST. This activity is the one that is most recognisable in the industry as it has been performed for many years. This includes traditional penetration testing such as black-box and white-box testing. SAMM also suggests performing more tailored testing based on test cases derived from the security requirements as well as source code analysis for information security related issues using automated tools.

Within this article through the introduction of OpenSAMM, I have highlighted the core areas in which we work with organisations as they address their issues around how to implement and operate information security into their software development processes. As software assurance is still a young discipline within a young industry such as information security, we believe that organisations can benefit greatly in embracing a framework such as OpenSAMM. This is in much of the same manner as how the BS7799 standard was embraced by organisations over the past 10 years to incorporate the fundamentals of information security into their business.

Matt Bartoldus is an information risk management professional with over 10 years of experience managing and delivering information security projects. Service delivery experience spans the scope of IT audit; security penetration and vulnerability assessments; regulatory compliance and information security governance consulting; policy and standard development; as well as security business transformation.

SNIFFING OUT THE CROOKS

Tom Ilube is CEO of Garlik, the online identity company which he helped to found in 2005. He is also former CIO of the internet bank Egg, and he spoke to Henry Tucker about identity theft, professionalism and building secure software.

Having left Egg, Tom said that Garlik's founders wanted to build another large-scale consumer-focused technology business and looked at two things: what was happening to consumers in the digital world, and what would be the interesting trends over the next few years? The main thing that struck them as interesting, he said, was the sheer amount of personal information that was appearing in the digital world about consumers.

What Garlik does is to look around the digital world, particularly focusing on those areas that are considered to be high risk - sites that trade personal information - scanning for personal information about the person who has commissioned the search, to see if it turns up in there.

Having done this they go back to the user and advise them of the information they've found and what they should do about it. The next step is often to get the data removed, but this isn't always possible. However, there may be other possible steps.

'One example I like to give is that banks often ask you for your mother's maiden name as a security word,' said Tom. 'You will discover that your mother's maiden name is publicly available on the internet along with everyone else's because the births, marriages and deaths records have been published.

'The bank doesn't care after all. So why give them the precise word that's on that list that anyone can see? So even when you can't get the source information removed, sometimes there are smart things that you can do.'

Having discovered data that's out there, sometimes Garlik can trace how it was compromised as it's obvious. However, often they can't do so at an individual level, but because they have seen so much information from so many similar sources, they can tell that various people are in the same situation.

'For example there was one case we came across where someone had used an online retailer to buy something and that site had parted company with its system administrator. It turned out that on leaving, he made a couple of changes that altered all of the details that were held in secure form on their server into open text and made them publicly available.

'Now this system administrator clearly did something wrong. Whether he committed a crime is not clear; he didn't use the information for personal gain, he was just really irritated at the way that things had come to an end and so put a tick box there where there wasn't one before.'

According to Tom, the problem now is companies don't appreciate just how valuable the personal information they hold is and that they are holding the information on trust for the people.

'The mindset in the bank traditionally has been that it is looking after the money but it's not their money. Companies though don't have the same sort of mindset when it comes to personal information. They feel that people have given them their personal information and they can use it to run their business. They just have the view that "we've harvested Tom's information, that's mine now and I can do whatever I like with it." Fundamentally though it still belongs to the individual and the company has a duty of care to look after it in the same way as if it were something else of value.'

Tom feels that if companies adopted the above mindset, the controls that companies would put around their use and their management of that personal information would get a whole lot tighter.

To back up this theory, Garlik did some research into the government, issuing 30 freedom of information requests across almost every major government department. It asked a series of questions about how they handle their information. Garlik asked if departments had ever been independently audited to see whether they comply with the Data Protection Act? None have.

'They are not required to have audits, but if they did and they don't cost a huge amount, but just said once a year: "We want our existing auditors to also check that we are compliant", you would just see people reacting,' said Tom. 'People would say: "The auditors are coming, let's have a look at our processes and procedures and so forth." And they don't do that at the moment.'

Another question they asked was to see if the information held by the organisation was correct. This is very important because if the information is wrong and decisions are being made about an individual based on it, how can they put it right? Garlik asked what policies and procedures these departments had in place to check that the information that they hold is correct and how a person can correct it if something goes wrong.

Garlik soon found out that hardly any of them have got these basic procedures in place. None of them keep statistics on the error rates in their information, none of them have budgets put aside for error correction.

Duty of care

'I think it comes back to the issue of companies and government and owners of big databases not treating these databases as things that they have a duty of care to look after. If the systems administrator in that company really felt "I've got a duty of care to look after this information" it would have some double checks in place and procedures when people leave so that someone else checks things. If data is lost, the organisation still has it, unlike when money is stolen, so they don't feel a sense of loss.'

When asked if embracing professionalism would help improve this situation Tom agreed.

'I think professionalism within IT security is really important. One of the challenges for us in the IT industry is to build systems that are secure by design. What often happens is the emphasis is on the facing functionality and at the end of the cycle the security guys get involved and have to almost patch up the security as best they can, given that it has already been built and you have business guys standing there saying: "Put it live, put it live."

He feels that if you can get security people involved in the design process right from the start and if you can get software engineers educated in how to build secure systems, it just becomes a part of what they do.

With this in place you would end up with the whole industry building much more secure systems and that feeds through into trust in an age when the mainstream consumer uses the systems. It's not just the professionals using the system, it's also about our parents and the man on the street, and so forth, who will be attracted to things that they can trust and trust comes through security, so security gives a competitive edge as well.

Systems built securely will avoid data breaches too. If breaches happen and a company deals with them very professionally, it has an impact on the business. People see security and the professional approach to developing software that results in secure systems as a constraint on a business, the speed at which they can move and get things done.

'What I say to people is look at it like the brakes on a Formula One car,' said Tom. 'The reason why Lewis Hamilton can go faster is because he knows that he is in control with those brakes, he can accelerate because he knows he can tweak those brakes and come round the corner. So those controls aren't stopping him, they are enabling him to go that fast.

'In Egg one of the things I did as CIO was to introduce an agile programming approach and agile development. Sometimes people can misunderstand that agile approach and say either you can have a professional waterfall approach or you can throw caution to the wind and go agile.

'But actually an agile approach is extremely professional as you get all of those people involved right from the start before anything has been created. I think we need to think broader about what is professionalism and not get stuck on professionalism always looks like this. The idea that we as an industry need to get more professional in what we do – I definitely agree with that.'

Tom also feels that a way to deal with the increasing amount of data loss is to implement laws similar to the data breach notification legislation in place in the US.

He believes that the UK and the EU will end up with similar data breach laws to the USA. He thinks, that part of the challenge is that, even in the US, in some states individuals are getting quite used to being told that their data has been breached and sometimes they start to tune out.

'I think what we don't appreciate with all these stories is that the information being lost is very useful to these sorts of people. If it goes missing today it's going to come back and bite the people and we're not really feeling the effects of it at the moment.'

What he thinks is going to happen if you project forward two or three years is that the amount of information about all of us will continue to grow. If you're on Facebook or LinkedIn, Google and all of these places where the information doesn't go away, then there will be times when people have talked about you or tagged you in photos, as well as information that the government has put out there about us, and documents that we have filled in that the local authority has put on its website. The information keeps flowing.

With this information, identity thieves have worked out that it's can be easier to hit individuals with a fraud than to hit an organisation. The only reason that they don't do it in a big way at the moment is that if you hit an organisation with a fraud and it works, you make more than if they hit individuals. But if they can hit a thousand individuals at the same time and obtain money from everyone, maybe that's a lower risk, easier way to do it, rather than via one big organisation.

This doesn't mean that all is already lost though. Tom feels that currently we have a window where we can recognise that personal identity is important and needs to be looked after and get that into individuals' mindsets, governments' mindsets and use some of the technologies that can help us in this area.

'If you leave it then I think it could get very difficult,' said Tom 'It's like Spam, if there were no spam filters then email would be unusable. Why we don't see it is because there is a whole industry battling that problem.'

Best Practices for Secure Software Development

Security attacks are moving from today's well-protected IT network infrastructure to the software that everyone uses – increasing the attack surface to any company, organisation or individual. Paradoxically, productivity-enhancing software that is embraced often invariably houses large amounts of sensitive data, both personal and corporate writes Mano Paul of (ISC)2.

The infamous release-and-patch cycle of software security management can no longer be the modus operandi or tolerated. A growing community of professionals, supported by the global information security professional certification body (ISC)2®, understand that escaping this vicious cycle requires a systemic approach.

Given below is a compilation of ten best practices for secure software development that reflect the experience and expertise of several stakeholders of the software development lifecycle (SDLC). These stakeholders include analysts, architects, coders, testers, auditors, operational personnel and management.

1: Protect the brand your customers trust

As cybercriminals evolve, so must the defenders. It's the defenders and their organisations that need to stay a step ahead of the cybercriminals as they will be held responsible for security breaches. Breaches leading to disclosure of customer information, denial of service, and threats to the continuity of business operations can have dire financial consequences. Yet the real cost to the organisation will be the loss of customer trust and confidence in the brand. Such a loss may be irreparable and impossible to quantify in mere monetary terms. Fundamentally, the recognition that the organisation is obligated to protect the customers should powerfully motivate the organisation in creating more secure software.

2: Know your business and support it with secure solutions

The answer to the question – 'Why were brakes invented?' could be answered in two ways, 'To prevent the vehicle from an accident' or 'To allow the vehicle to go faster'. Similarly, security can prevent the business from a crash or allow the business to go faster. One must work with a thorough understanding of the business, to help in the identification of regulatory and compliance requirements, applicable risk, architectures to be used, technical controls to be incorporated, and the users

to be trained or educated.

3: Understand the technology of the software

A thorough understanding of the existing infrastructural components such as: network segregation, hardened hosts, public key infrastructure, to name a few, is necessary to ensure that the introduction of the software, when deployed, will at first be operationally functional and then not weaken the security of the existing computing environment. Understanding the interplay of technological components with the software is essential to determine the impact on overall security and support decisions that improve security of the software. Further, when procuring software, it is vital to recognise vendor claims on the 'security' features, and also verify implementation feasibility within your organisation.

4: Ensure compliance to governance, regulations and privacy

An industry that is not regulated is today an exception to the norm. Governance, risk and compliance (GRC) is a means to meeting the regulatory and privacy requirements. One must understand the internal and external policies that govern the business, its mapping to necessary security controls, the residual risk post implementation of security controls in the software, and the compliance aspects to regulations and privacy requirements.

5: Know the basic tenets of software security

When it comes to secure software, there are some tenets with which one must be familiar: protection from disclosure (confidentiality), protection from alteration (integrity), protection from destruction (availability), who is making the request (authentication), what rights and privileges does the requestor have (authorisation), the ability to build historical evidence (auditing) and management of configuration, sessions and exceptions. Knowledge of these basic tenets and how they can be implemented in software is a must have while they offer a contextual understanding of the mechanisms in place to support them. Some of these mechanisms include encryption, hashing, load balancing and monitoring, password, token or biometric features, logging, configuration and audit controls, and the like.

6: Ensure the protection of sensitive information

Any information upon which the organisation places a measurable value, which by implication is not in the public domain, and would result in loss, damage or even business collapse, should the information be compromised in any way, could be considered sensitive. While it may be easy to identify the sensitivity of certain data elements like health records and credit card information, others may not be that evident.

One must consider data classification and protection mechanisms against disclosure, alteration or destruction. Data classification is the conscious decision to assign a level of sensitivity to data as it is being created, amended, stored, transmitted, or enhanced, and will determine the extent to which the data needs to be secured. Software that either transports, processes or stores sensitive information must build in necessary security controls.

7: Design software with secure features

When someone is exclusively focused on finding security issues in code, they run the risk of missing out on entire classes of vulnerabilities. Security issues in design and other concerns, such as business logic flaws need to be inspected by performing threat models and abuse cases modeling during the design stage of the software development lifecycle. Threat modeling, an iterative structured technique is used to identify the threats by identifying the security objectives of the software and profiling it. Attack surface analysis, a subset of threat modeling can be performed by exposing software to untrusted users.

8: Develop software with secure features

It is imperative that secure features not be ignored when design artifacts are converted into syntax constructs that a compiler or interpreter can understand. Once developed, controls that essentially address the basic tenets of software security must be validated to be in place and effective by security code reviews and security testing. This should complement and be performed at the same time as functionality testing. Definition of the scope of what is being reviewed, the extent of the review, coding standards, secure coding requirements, code review process with roles and responsibilities and enforcement mechanisms must be pre-defined for a security code review to be effective, while tests should be conducted in testing environments that emulate the configuration of the production environment to mitigate configuration issues that weaken the security of the software.

9: Deploy software with secure features

Secure deployment ensures that the software is functionally operational and secure at the same time. It means that software is deployed with defence-in-depth, and attack surface area is not increased by improper release, change, or configuration management. It also means that assessment from an attacker's point of view is conducted prior to or immediately upon deployment. Software that works without any issues in development and test environments, when deployed into a more hardened production environment often experiences hiccups. Post mortem analyses in a majority of these cases reveal that the development and test environments do not simulate the production environment. Changes therefore made to the production environment should be retrofitted to the development and test environments through proper change management processes.
Release management should also include proper source code control and versioning to avoid a phenomenon one might refer to as "regenerative bugs", whereby software defects reappear in subsequent releases. The coding defect (bug) is detected and fixed in the testing environment and the software is promoted to production without retrofitting it into the development environment. Further, vulnerability assessment and penetration testing should be conducted in a staging pre-production environment and if need be in the production environment with tight control.

10: Educate yourself and others on how to build secure software

As Charles Dickens once eloquently said: 'Change begets change.' When one who is educated in turn educates others, there will be a compound effect on creating the security culture that is much needed—to create a culture that factors in software security by default through education that changes attitudes. IT security is everyone's job.

Crashing cars and firewall management

The similarity between the two might not be obvious but they can set off a chain reaction according to Calum Macleod, regional manager, Tufin Technologies.

With all the doom and gloom of the past few months and billions of whatever currency you like being poured into the economy I have to report on a ray of hope. I think my son may have hit on the solution completely inadvertently. He's not a renowned economist, just an honest, hard working car mechanic.

However, having written off the fifth car in the last three years, although credit where it's due, this time it was his fiancée that managed it, not only is he trying to save the motor industry single handedly but at the same time his insurance premiums have reached a level where he may be also saving the financial sector.

Not only that, but out of sympathy I've had to break open the reserves and help finance number six, which of course means that what money I had left is now circulating.

But what, may you ask, does this have to do with IT and security come to that? Actually quite a lot because his latest accident triggered a chain reaction that we're all too familiar with.

First, a lack of risk assessment resulted when according to his fiancée 'a woman driver decided to stop on orange' with the result that she ploughed into the back of the car. Mind you had the mechanic bothered fixing his brakes, as everyone was telling him to do, it all might have been avoided. And as is so often the case in IT security, improper risk

assessment can have disastrous consequences.

Not enforcing information security policies or firewall policies can very often result in failed audits, network breaches and other such things.

Firewall rules

Second, it had major business continuity impact. Having no car meant having to borrow somebody else's car. Everybody was impacted. A very common problem in many organisations is the impact on day to day business because of errors being made in translating service requests into structured firewall changes, or failing to adhere to information security policy, or placing firewall rules where they should not be, brings everything to a grinding halt. Third, the failure to deal with the risk resulted in a problem, with the result that the financial impact on the family organisation was significant.

I'm not saying the accident would not have happened but had the brakes been working it might have resulted in what became a 'right-off' being no more than a small dent.

Bottom line failure to deal with the risk in order to save money eventually ended up costing a lot more than it should have. So what should you do? Use automated risk assessment tools – fix the brakes.

Risk assessment

One of the key reasons why risk assessment is not done is simply that it is extremely time consuming if it is done manually. When I ask companies the question, the responses vary from 'we have never done' a risk assessment to 'so far we've got away with it because the auditors have never asked.'

Additionally it is surprising even among financial institutions that auditors are not addressing this problem. This is likely to be due to the fact that they do not know what to look for. Relying on specialist consultancy companies to do this job for you can also be a very hit and miss affair because you are at the mercy of a consultant who may or may not have the necessary skills to do this. And in any case if they haven't got the right tools the chances are they're no better than anyone else.

Best practice

The only effective way to really assess if your firewalls are protected is to use tools that are able to examine your firewall configuration based on known best practices. Additionally, the better tools allow the firewall administrator to address new vulnerabilities in real time. Since this process is fully automated it takes the manual, subjective approach away from this task and it ensures that you can analyse in minutes what would normally take weeks or months to do manually. And this has to be a continuing process.

Communicate with the business and know what are your business critical applications. It won't be a big surprise but many IT administrators and firewall administrators do not know which applications are business critical. The result frequently is that either rules are left in firewalls because no one dares touch them, which in turn results in poor firewall performance. The other situation that often occurs is that rules or services are removed because they do not appear to be used. Again the problem is frequently due to the fact that manual processes are used to examine usage and very often services can be unused for months simply because the applications that use them are not run on a regular basis but may be business critical.

Policy management

Again the only effective way to ensure you avoid these mishaps is to use technology. Firewall policy management (FPM) technology enables an organisation to define business critical applications so that any changes which impact these applications can be identified quickly. In fact some tools enable you to model scenarios before making changes. The modeling allows you to identify if a change will impact business continuity so that you can avoid making the errors in the first place.

Another key use of FPM tools is being able to translate business requests into actual changes. In a recent meeting a customer told me that they spent two days trying to activate a service for a client because they were not able to identify that changes were required on two firewalls to enable the service. An FPM tool that provides 'What If' capability will ensure that all necessary changes are shown before implementation is necessary.

Rule usage analysis is also a major problem without the proper tools. Administrators can very often take days to analyse a single rule because as rules move in the rule base, without automated tracking tools it is virtually impossible to follow the rules and their contents in a large rule base.

Fix the brakes

Choosing to deal with the risk or leaving it in the hope that it doesn't happen to you is a choice you make. Not dealing with it is hoping that your colleagues don't make mistakes. So like my son, if you're going to let somebody else 'drive' your firewall, you'd better be sure that the 'brakes' are working.

www.tufin.com

Control or manage?

The four pillars of information security are: confidentiality, integrity, availability and compliance. A good security manager will ensure a balance of these components across the enterprise, based on materiality and risk. The IT side of information security can be controlled pretty absolutely, but the people side can only be managed because people have free will and can choose whether, or not, to follow a particular policy, standard, or procedure.

This difference between control and management needs to be understood when balancing the absolute against the discretionary in any security structure, process, or mechanism. People are both the strongest and weakest components in the security paradigm in that they have the ability to both detect security weaknesses and also to create them.

Technology does not deliberately attempt to manipulate a process, although it may sometimes feel like that. People on the other hand have the ability to make poor systems work and good systems fail. So if security is only as strong as its weakest link, then once the security process is in place it is only likely to vary at the behest of a person. Technology may fail but it is not manipulative, so if I find an error in some software I know with a great degree of certainty that the error is there as a result of either human failure or deliberate manipulation.

Which brings me neatly to the self assessment income tax software supplied by Her Majesty's Revenue and Customs (HMRC). I use this software to complete my tax declaration to HMRC and as they provide it I expect it to be suitable for its intended purpose. Not so. This year I detected three quite significant errors in the software. Initially it did not take any notice of any declared foreign income, which was quite surprising in view of the government's publicity campaign to persuade us to declare such income. This was, however, quite swiftly rectified. Next, the PDF copy of what I was submitting did not agree with what I had input. This is quite serious as it means that I do not have a true record of what I have submitted. I have received a fulsome apology from HMRC for this, but the fault still remains un-rectified since I first reported it last September. Third, the actual tax year for payments made on account is incorrect. I have reported this too, but still have not received an acknowledgement despite sending three reminders.

I used the Freedom of Information Act to ask how many software errors had been detected in the previous twelve months and how many remained outstanding? A commendably swift response revealed that 70 software errors had been reported of which 25 were still outstanding. Remember, this is live software being used by citizens to declare their earnings. During my correspondence with the help desk I asked that as I could not be certain that what I had entered was what they received, did they have a 'work around' for me. Yes, they responded, I could submit a paper return.

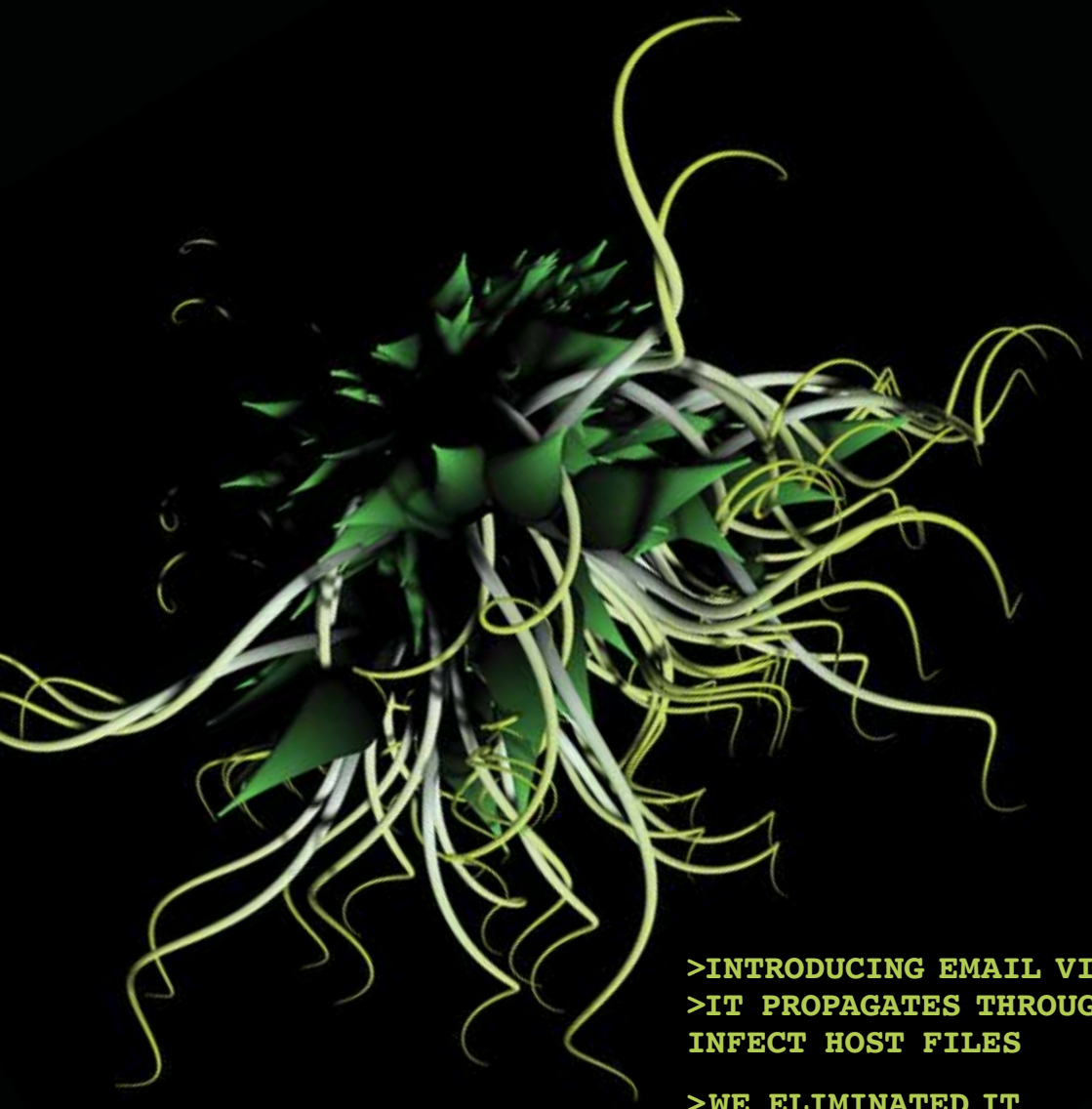
I pointed out that the deadline for paper submissions had past. Never mind, they responded, just attach a note explaining things and perhaps my local tax office would look kindly on me. This altruism was somewhat tarnished by the reminder that a late submission made me liable to a £100 fine, but it was unlikely that I would be charged interest on any late tax that I owed! So this is how it goes. You use software provided by the government at your peril. It may contain known errors, but you will not be informed you of these.

Their testing process is so poor that the software enters production with numerous errors. You have a duty to submit an accurate and timely tax return, but cannot be sure of what you are submitting as the PDF copy provided by their software does not agree with what you have entered.

When you raise issues you are told to drop the technology and revert to pen and paper, but what if you cannot drop the technology? If your company submits PAYE information to the government then there is no option other than to use the technology as paper returns are no longer permitted. One hopes that the PAYE testing and change management procedures are more robust than those used by their self assessment colleagues.

My letter to the chancellor on these issues remains unanswered, but I suspect that he has more pressing concerns at the moment than the integrity of his tax calculation software.

John Mitchell is Managing Director of LHS Business Control, a corporate governance consultancy. He is currently a member of the BCS Specialist Groups Executive Committee.



>INTRODUCING EMAIL VIRUS NETSKY
>IT PROPAGATES THROUGH EMAIL TO INFECT HOST FILES

>WE ELIMINATED IT
>AND IT FELT GOOD

>WE DISCOVER & DESTROY THOUSANDS OF UNIQUE VIRUSES EACH MONTH
>THE ANTI-VIRUS COMMUNITY RELIES ON OUR ADVANCED INTELLIGENCE
>WE HAVE A 100% GUARANTEE AGAINST ALL VIRUSES
>VISIT MESSAGELABS.COM/THREATS FOR A FREE TRIAL

MessageLabs | Now part of Symantec

IMAGE GENERATED USING SOURCE CODE FROM EMAIL VIRUS NETSKY


UNIVERSITY OF
OXFORD
part-time study
network security
trusted computing
systems design
security processes
people and security
msc in software and systems security
www.softeng.ox.ac.uk/security