



OVUM CONSULTING

Developing secure NGN infrastructure

Ovum research for Codenomicon

Reference Code: CYIT0108

Publication Date: September 2011

Author: Graham Titterington

THE OVUM VIEW

Next Generation Networks (NGNs) provide a rich range of IP-based services for telecommunications operators, including voice, data, video, TV and messaging. The use of IP protocols as the foundation of NGNs gives great flexibility, but also exposes the networks to all the security threats found on the Internet. Operators have to address a set of issues that is very different to those on the tightly constrained but functionally limited legacy networks. NGNs do not have a strong separation between signaling and control channels, and the payload channels. Denial of service attacks, toll fraud, information theft, and user privacy threats are real in NGNs. Networks also connect to user endpoint devices and other networks, where the NGN operator cannot control the status and "hygiene" of the equipment. The complexity of the network makes it difficult to secure.

NGN component vendors and operators are working together to address security challenges. They are using secure development processes, supported by a large range of security testing tools. It is essential that they continue, and strengthen, these initiatives. NGN services are business critical, and in some cases safety critical, and people expect the same levels of reliability that they associated with conventional telephony.

Security assurance can be delivered by using a range of methods in conjunction with each other, including secure development and deployment processes, peer reviews at all stages in the development life cycle, "static" analysis of code and configuration, and "dynamic" testing of components. Security testing needs to examine the communication layers below layer 7 of the OSI stack, while security testing of applications examines functional correctness at layer 7.



"Fuzz testing" is particularly suitable for testing the reliability and robustness of components that handle NGN protocols. The fuzz testing tools that are now available can make a major contribution to ensuring the security of NGNs.

RESEARCH METHODOLOGY FOR THIS STUDY

This research is based on interviews with nine NGN service providers and vendors serving this market. The interviews were conducted with technical experts in each of the companies, and in some cases second interviews were arranged to extend the scope of the discussion. The companies all currently use Codenomicon's fuzz testing security products.

THE SECURITY ISSUES RELATING TO NGNS

Evolution of services

The move to IP NGNs in the telecommunications world is happening slowly, with customer sentiment governing the speed and obligations to maintain PSTN services continuing for the foreseeable future. The risks that arise from the move are gradually emerging as operators offer more NGN services, and security incidents are detected. The process of moving all telecommunications services may be complete by 2025 but may take longer. What is certain is that network operators and their suppliers have to address the security issues now and design security in to all future products and services. The speed of the transition does give network operators and their suppliers an opportunity to understand the issues in NGNs fully.

Industry concerns

Operators' perception of the security risk today varies, with those operators that already have a broad portfolio of IP services being most aware of the threats. The greatest concern is with attacks coming from the Internet through the user's devices to the network core, with considerable concern that these might attack the NGN signaling protocols. There is almost as much worry about the risk of attacks coming from other operator networks through interconnection interfaces. Operators also expressed anxiety about attacks aimed at their users, privacy, and intellectual property held within their services, and of attacks on legacy TDM/PSTN networks through the NGNs - but these were at a lower level than their concerns about attacks on the NGN core.

The specific consequences that operators fear include:

- Denial of service attacks (DoS) and Distributed Denial of Service attacks (DDoS), particularly if it disables a service
- Theft of personal data
- Reputational damage

- Toll fraud
- Legal action from the regulator resulting from a security failure.

Attacks can come at any layer in the NGN, including at the management layer where unauthorized access could allow billing fraud. As mobile devices become more sophisticated we are seeing a migration of intelligence to the edges of the network where it is more vulnerable. The objectives of delivering secure, robust and reliable services are interconnected because a failure in any of these aspects has consequences for the other two aspects.

Complexity

The changes in the network operators' world is characterized by increasing complexity in several dimensions:

- Technological capability
- Service offerings
- The supply chain, with outsourcing and sub-contracting at every stage
- The regulatory environment.

Complexity is the enemy of security, and it is becoming increasingly difficult for a player in the industry to fully understand its dependencies and exposure to threats. There is a need for every player to be able to demonstrate its security credentials, and to be able to understand the level of risk they are exposed to from other players.

Technological complexity also directly increases the risk of a failure, as human made errors in software and hardware become more common in larger and more complex systems. At the very least the number of errors is likely to increase proportionately with the size of the software code, and historical evidence from software development studies indicates that the number increases more rapidly than this unless action is taken to improve code quality.

Service levels and costs

A security incident can be regarded as an extreme example of a failure to meet service levels, and so security protection is part of service management. In particular, there is a need to allocate network resources between the range of IP services offered on a NGN. When this partitioning of resources is enforced it can limit the impact of a DoS attacks to one service.

Formal quality of service commitments are often given by network operators to commercial customers, but rarely to consumer customers. Failure to meet the Service Level Agreement (SLA) results in the operator making a payment to the customer as specified in the particular contract.

This is normally calculated as a refund of a portion of the service charges while at least one major



operator gives additional compensation, although this operator still calculates compensation on the basis of service lost rather than on the business consequences of the failure. It is an established practice for operators to prioritize types of traffic where QoS is most critical, such as prioritizing voice over data, but with the growth of bandwidth hungry and QoS sensitive traffic such as video, there is an emerging need for a more sophisticated allocation system to maintain acceptable standards across the service spectrum. Many operators still rely primarily on over-provisioning bandwidth, but this strategy may not be practical with the growing range of NGN services. Clearly more thought is needed on bandwidth allocation, and this should be done in conjunction with defending against DoS/DDoS attacks.

Most operators have not developed a means of accounting for the full cost of service downtime to either themselves or their customers. They keep statistics of faults and disruption, but the financial implications of these are not generally monitored. Most vendors also track defects but do not quantify the associated costs. The cost of a defect to a vendor in an existing product includes the direct cost of responding and issuing a patch, possible contractual refunds to operators, technical support costs, and the "opportunity cost" of diverting resources from new development.

Sources of threats

Network operators do not trust their industry colleagues to feed them clean communications! All the operators that we interviewed said that they applied different levels of trust to each operator with which they interconnected, and that none of them were fully trusted.

PCs are vulnerable to all the Internet malware threats and these could be carried onto the NGN. However the main concern of NGN operators with regard to PCs connected to their network is the risk of botnets on the PC contributing to a DDoS attack.

The security landscape of mobile handsets is becoming more like that of PCs as they become more powerful, and the Android can be regarded as a mobile Linux device.

2G/3G stacks such as GNU Radio are not yet widely deployed, but the fear is that they may allow hackers to use them to impersonate networks to end user devices, and vice-versa, creating more opportunities for data identity theft, and for launching DDoS attacks.

Similarly open application APIs on end user devices add to the complexity of the overall system configuration and make it more difficult to identify the threats that may exist on the endpoint. Set top boxes present a particular threat to the intellectual property that may be held on them.

Most operators believe that the move to the IPv6 protocol across the Internet is likely to increase the threat level in the short term due to network administrators being unfamiliar with it, and to a shortage of security devices designed for this protocol. Although it is inherently more secure than the older IPv4 protocol, there was also concern that the large address space, providing plenty of



temporary IP addresses and multiple addresses per device, will make reputation management more difficult in the longer term.

Security as a factor in the market place

Security concerns are an important factor constraining the business development of NGN operators. Security is therefore an enabler in the development of the business. Half the operators we spoke to have restricted their service offerings because they considered the security risks of a proposed service could not be reduced to an acceptable level. In other cases the operators have delayed the introduction of services to provide time for security enhancements to be delivered.

Over 80% of the operators we spoke to thought that their security efforts could be used as a competitive differentiator in the market place, at least for their commercial customers.

On the other hand security is also a factor when operators are choosing vendors to supply them. The majority of operators select products after performing a security assessment, and others rely on external product certification, but one admitted that price was a more significant factor in the selection. Vendors reported that the operator interest in security testing was spreading from traditional markets in North America and Western Europe to emerging markets.

From the vendor perspective, our respondents told us that their security enhancing work is driven by their internal standards, inspired by customer demands and industry regulation. The number of customers demanding proof that they have used secure development processes is growing. They satisfy this demand either by obtaining third party certification of their products or by sharing technical analysis or testing results.

HOW BUSINESSES ARE WORKING TO REDUCE THESE RISKS

Where to protect

Operators believe that they need to protect all their boundaries:

- Internet connections
- Customer access points
- Interconnection gateways with other operators
- Administrator consoles.

They place particular importance on protecting signaling channels.

Legacy networks can best be protected by maintaining a degree of separation from NGNs, based on well considered policies. This includes blocking IP access to legacy administration and tightly monitoring connections to drop malicious traffic.



Operator security initiatives

Operators are taking different steps to address security concerns, depending on the range of NGN services that they offer and their procurement strategy, for example how much development do they do in-house.

At an operational level they are deploying several types of security products within their networks including Session Border Controllers (SBC), firewalls at the network edge and in the DMZ, Information Gateway Services (IGS), Intrusion Prevention/Intrusion Detection Systems (IPS/IDS), Traffic Control Systems, and Virtual LANs (VLAN) to separate traffic on the control plane.

However technology alone cannot protect the operational network. Good management of the network is important, including:

- Proactive monitoring of the backbone and carrying out reactive measures against the source of a DoS attack. Once a DoS or DDoS attack is confirmed its packets can be dropped
- Separation of the management network from the customer service network. Applying state aware checks to protect signaling
- Rate limiting the traffic on each service
- Checking the origin of each packet to detect DoS/DDoS attacks
- Being aware of the reputation of other networks
- Monitoring live systems so that systems can be made more resilient in future
- Having a well-defined incident management process
- Maintaining strong access controls to administrator consoles and monitoring all administrative actions
- Patching all systems promptly and having good patch management.

In addition operators are also becoming increasingly aware that they have to build in security when they design new services and new infrastructure. This includes conducting architecture reviews and security audits before deployment, and encouraging vendors to carry out security testing. Particular attention must be paid to verifying the performance and correct operation of all protocols when dealing with all possible workloads, including malformed traffic.

Developing secure products and systems

Organizations should have a Security Development Lifecycle (SDL) to structure their development processes. We found that all the vendors and operators have an SDL except for those operators that outsource all their software development. In this case security is maintained by testing at the



procurement stage. Most organizations have developed their own SDL, but some have modeled them on the Secure Software Development Lifecycle of major vendors such as IBM or Siemens.

It is important to consider security right at the start of the development process, both to make it effective and to minimize its cost. The initial phase should include a threat analysis and a definition of security requirements, and risk assessments should be conducted throughout the lifecycle. Each document in the design process should be reviewed from a security perspective. Code should be produced in accordance with secure coding best practices and verified by submitting the code to a code analysis tool. Products should be hardened by removing unwanted functionality.

The development lifecycle should incorporate best practice at every stage, and should include mechanisms to continuously improve the process itself.

A large part of NGN system development is outsourced and security management in an outsourced environment is difficult. The supply chain is invariably complex as both outsourcers and vendors buy in components from their suppliers. As far as possible, operators should try to understand their supply chains and identify any weak players. For the most part operators rely on testing the delivered code when they buy in products. We look at the security testing process in the next section. Operators can help themselves in the long term by providing feedback to their suppliers and encouraging them to raise their security standards. We found vendors that admitted that customer pressure had encouraged them to adopt more secure development processes.

Formal standards do not play a major role in the development of NGNs. We found localized use of the following security standards: ISO 27000/27001, Common Criteria, CMMI, ATS, NIST Special Publication 800 series, and OWASP standards, but no over-arching requirements for particular standards. Vendors expressed a desire to see a rationalized set of standards with organizations such as the ITU and IETF taking the lead.

SECURITY TESTING

The range of testing approaches

Security testing plays a crucial role in delivering secure systems and products. It is so central to the delivery of reliable services that we found that NGN operators who had previously outsourced security testing are now bringing the testing back in-house. There are three main categories of "test" available:

- Manual inspection and audit
- Static Analysis Security Testing (SAST)
- Dynamic Analysis Security Testing (DAST).



We found that most vendors and operators are using all three approaches, and many types of tool within each of them. Both vendors and operators thought that DAST was more critical than SAST, but that the two approaches were complementary and contributed different benefits to the quality assurance process. We include fuzz testing, conformance testing, message content stability testing, and simulators in the DAST tool category.

Many of the security testing requirements of NGNs are specific to the sector, and in particular there needs to be a strong emphasis on ensuring that all interaction protocols work correctly, even when under attack. NGN systems are characterized by large numbers of interacting entities. As a result of this we found a wide range of security testing products in use, ranging from proprietary specialist products, through many free and open source testing product, to in-house specific tools.

In-house developed testing tools are expensive to maintain and often difficult to use as they are developed for a small user base and tool development budgets are restricted. Developers use them because commercial tool suppliers do not meet the specialized needs of the industry. Open source products provide a platform that can be customized to specific requirements, as well as relief for the project budget! Many of these tools have achieved the highest ratings in their niche, making them suitable for the critical nature of NGNs. However developers would like to have a proprietary product if it precisely meets their needs.

Within the proprietary testing tool sector, Codenomicon stands out as the most important tool for NGN developers and operators. It provides a product that automates "fuzz testing". This specifically addresses the need to thoroughly test protocols without having to invest large amounts of effort on developing test cases.

Fuzz testing

Fuzz testing takes the definition of the way that a protocol should behave and generates a suite of test cases around this incorporating large numbers of mutated and ill-formed tests. These test the stability and efficiency of the products that handle the protocol. The tests address the core issues in NGN security. The tool also automates the management and execution of these test suites. The testing process is so massive that it is impractical to perform comprehensively without automated support.

Codenomicon supports three levels of test case design:

- Random test case generation
- Template-based, where a captured stream of legitimate traffic is used as the basis for fuzz testing
- Fuzz testing based on a formal model of the protocols and their interactions.



We found operators working at all three levels, and testing at all appropriate interfaces within their systems. Vendors were more likely to base their testing on a formal model of the component under test.

Why did industry players choose their particular security testing strategy?

Operators have to be sure they can satisfy their requirements for performance, scalability, service level, quality, and functionality. They need to be able to focus testing resources on components where the current level of assurance about these requirements is lacking. Much of their infrastructure and software is bought in and generally has to be tested as a "black box".

Vendors are influenced by similar factors, as well as pressure from their operator customers to deliver assurance in a form that meets the operator requirements.

Both vendors and operators seek the maximum return on their investment in security testing within the limits of the technology that is currently available.

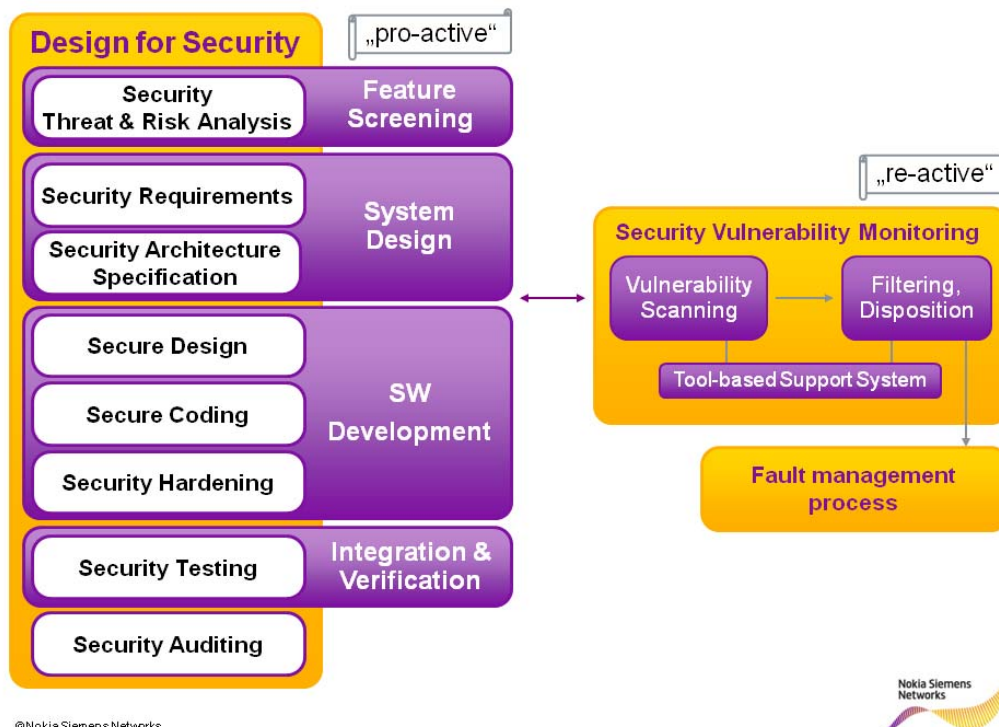
CASE STUDIES

Nokia Siemens Networks

Nokia Siemens Networks (NSN) provides a wide range of products for telecommunication networks. It believes that vendors have a responsibility to protect against the threat of an Internet-based attack on these networks and that the risk is serious, although its true extent has not yet become apparent. It is working to protect against all the credible types of attack because any attack can damage the operator's business.

For developing secure products, NSN has a process called "Design for Security". The broad outline of this process is shown in Figure 1. It includes threat analysis, gathering security requirements and defining a security architecture. Code is written according to industry best practice from a secure coding perspective. Automated code analysis tools are also used to identify and eliminate software bugs which might lead to vulnerabilities. Security testing is done alongside functional testing at the product integration stage. NSN integrates products from third parties in its deliverables. These components are additionally subjected to vulnerability scanning.

Figure 1: Nokia Siemens Networks Product Security Management Process



Source: Nokia Siemens Networks

NSN uses fuzz testing in both the security testing and security audit phases of a project. It was involved with pioneering work in this process as early as 1999. It has found this technique to be particularly useful for delivering robust systems that continue to function correctly in the face of an IP-based attack. NSN uses the Codenomicon suite of products for most of its fuzz testing and uses the Codenomicon test cases for testing the common protocols. This provides comprehensive testing without expending massive effort on test development. However NSN also develops systems using some less common protocols, including proprietary protocols, that are not supported by "out of the box" test scripts, and here it has to develop its own test cases.

The Codenomicon suite provides for useful instrumentation of the tests so that testers can, for example, see the state of the system under test following a failed test and relate this to log files, thereby gaining an understanding of any error conditions that are revealed. The tool management layer allows selective re-running of tests, for example to retest only cases that previously failed.



Telcordia

Telcordia builds NGN products itself, running large software development projects, as well as working for service providers around the world assisting them with their procurement of systems from other vendors. It is this second role that we will examine here.

Telecommunications providers used to focus on conformance to rigid telecommunication protocols, but the move to IP has made the protocols less rigid. The pace of change is rapid.

Denial of service attacks are a major concern for Telcordia's clients, as maintaining availability is critical in telecommunications. This concern is directly addressed by fuzz testing. However fuzz testing should be seen as just one part of a broader quality assurance process.

Telcordia tailors its security testing activities according to the type of product under development. The range of equipment types and application domains is too broad to use a single approach across the range. The testing regime has to react to the increasing threat level that we see in NGNs. Telcordia therefore has a comprehensive set of security testing tools, including home grown tools. These address testing of hardware, software, firmware, and communication protocols. Telcordia is looking at all aspects of a system component, and the interaction between components. Components can be PCs, laptops, set top boxes, network equipment, or service provider infrastructure. The component may provide services in the many areas such as telephony, entertainment, transport, and telematics.

Telcordia started using fuzz testing because service providers asked it to. Its use is increasing. It can be applied during the development phase before the vendor delivers the product, in which case Telcordia works with the development teams, or fuzz testing may be carried out at the end of the project as part of the quality assurance checks. Fuzz testing concentrates on communications interfaces and Telcordia mostly uses it for network products, rather than for endpoint devices. Historically fuzz testing started on internal interfaces within systems, and its use later extended to the external system interfaces where Telcordia is now deploying it. Before the Codenomicon tool was available, Telcordia was dependent on more extensive costly manual testing.

To conduct fuzz testing Telcordia mainly uses the test data that is supplied with the Codenomicon tool. This includes a library of attacks. Telcordia has access to all existing Codenomicon test libraries. For protocols that are not covered by these scripts, Codenomicon provides a capture tool to collect traffic from a system under test and automatically build fuzz tests by mutating the correct traffic packets.

In the IPTV domain, the testing requirement is usually specified by the content provider. Fuzz testing can be performed on alpha and beta product versions or on final version.

The Codenomicon tool automates fuzz testing, but more importantly it provides an analysis of what is going on inside the application when "garbage" is thrown at it. This addresses the increasing

threat of information theft within the systems. Codenomicon's analysis reports identify the issues and the internal environment so that the vendor can easily recreate the error conditions.

Huawei

Huawei is particularly concerned with the following threats in NGNs:

- The spread of hacking and information theft from the Internet to NGNs
- DoS and DDoS attacks on services, coming from the Internet and user terminals
- Attacks on service logic resulting in denial of service, user impersonation, and toll fraud
- Viruses and other malware in user terminals that damages services or commits fraud against the user.

Huawei has developed a security policy for its product development to protect against these threats. It models threats, designs security into its products, and adopts secure coding best practice. It tests its products using advanced methods and tools, and uses third party certification against standards such as Common Criteria. Huawei works with its operator customers to enhance their security design, configuration, and regular inspection. Its management policies include a security procurement policy, security deployment management, and security quality audits.

Huawei contributes extensively to industry standards bodies and is active in most of the relevant bodies.

Huawei has been developing its product development process since 1997. It introduced an Integrated Product Development process with help from IBM, and a Capability Maturity Model (CMM) process for software development where it has achieved level 5 maturity. Huawei has developed a Security Development Lifecycle based on SEE-CMM and Open SAMM.

Security testing is performed in all the phases of Huawei's SDL:

- Security threat modelling in the design phase of products, in order to decrease the security risks. Secure coding is mandatory in the development phase
- Security testing is required in all stages of the development process (UT,IT,ST,SDV,SIT and SVT)
- Before delivery of the product, the security laboratory working as an internal third-party certification agency, independent of product lines, will evaluate the security of the product. It refers any issues back to the product team to resolve
- Parts of products will be tested by third-party certification organizations outside Huawei, such as Common Criteria certification
- Finally security testing is performed by operators as part of the acceptance process.



Huawei uses many well-known assessment products as it places high importance on software quality.

Huawei uses automated testing products extensively. HUTAF (Huawei Unified Test Automatic Framework) is the principal automated testing framework and platform. It includes test management, performance level testing, security testing, and reliability testing.

Huawei also purchases commercial testing tools and services. Codenomicon's fuzz testing tool is an important component in this portfolio.

THE VALUE OF FUZZ TESTING

NGNs deliver information and services through various mediums:

- Land line and wireless
- VoIP
- Internet
- IPTV video
- Triple play
- TV, radio
- Smartphones

as well as providing support for legacy technologies.

NGN systems are complex and their complexity continues to grow, for example through the integration of IPv4 and IPv6. Complexity is the enemy of security and so it is important that NGN components are thoroughly tested, incorporating interoperability checks and quality assurance.

Security problems can include DoS/DDoS attacks, inappropriate access to content, impairment of quality of service levels, and access to the control level of the protocols. We need to test for features performance and robustness (i.e. resistance to unexpected or malformed data).

Fuzzing is a process for intelligently and automatically generating valid and invalid message sequences to see if the system breaks. There could be millions of such sequences to test and so automated test development, management and execution are essential. It is also important to have a test design tool or strategy that designs an intelligent and efficient test set, rather than an extensive randomly generated suite of tests. The available tools can be differentiated by their approach to test generation embodied in their fuzzing algorithms.

Both vendors and NGN operator can cut costs by a systematic approach to security testing, reducing the need for in-service patching.



Fuzzing does not need testers to have access to the source code, and so it can be used on third party software. We have seen that this is important because the industry is characterized by complex supply lines and many components have to be tested as "black boxes".

We have seen how Huawei uses extensive automated testing, in which Codenomicon's products play an important role. Telcordia particularly values the detail in the test reports because it helps developers to remediate issues that its tests identify. Nokia Siemens Networks uses fuzz testing both as a testing strategy and an audit process. It finds the approach ideally suited to the complex and extensive protocols found in NGNs.



RESEARCH FINDINGS

APPENDIX

Author

Graham Titterington, Principal Analyst, Information security, Ovum

Graham.titterington@ovum.com

Glossary

DoS attack

A denial of service attack is an attempt to prevent a computer system from performing its intended purpose. It is often perpetrated by swamping the system with unwanted traffic, but it can also be executed by destabilizing the system in other ways to force it into an irregular state.

DDoS attack

A distributed denial of service attack is a DoS attack in which the attackers use a battery of devices to deliver the DoS attack.

Fuzzing

Fuzzing, or fuzz testing, is a technique in which invalid, unexpected, or random data is input to a computer system so that the system responses can be monitored and evaluated. It is particularly used as part of the security testing process.

NGN

Next Generation Networks are advanced networks, usually based on the IP protocol, that are designed to provide many types of service simultaneously, such as voice, data, and video services.

OSI stack

The Open Systems Interconnection model (OSI model) derives from the work of the International Organization for Standardization, and provides seven abstraction layers by which the communications functions in a network can be defined in a logical, implementation independent, way. Each layer in the model provides services to the next layer above it.



Toll fraud

Toll fraud describes several types of financial attack on telecommunications services, designed to defraud either the network operator or the users of the network.

Disclaimer

All Rights Reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher, Ovum.

The facts of this report are believed to be correct at the time of publication but cannot be guaranteed. Please note that the findings, conclusions and recommendations that Ovum delivers will be based on information gathered in good faith from both primary and secondary sources, whose accuracy we are not always in a position to guarantee. As such Ovum can accept no liability whatever for actions taken based on any information that may subsequently prove to be incorrect.