



Codonomicon whitepaper:

Securing Next Generation Networks by Fuzzing Protocol Implementations

- Anna-Maija Juuso, Tero Rontti & Juha-Matti Tirilä -



- 1** Introduction
- 2** Vulnerabilities Enable Attacks
- 3** Reactive Defenses Are Not Enough
- 4** Discovering Vulnerabilities with Fuzzing
- 5** Vulnerabilities Are Security, Quality and Interoperability Issues
- 6** Fuzzing in the ITU-T X.805 Security Architecture
- 7** ITU-T Standard Y.2700
- 8** NGN Network Interfaces
- 9** Critical Interfaces in NGN
- 10** NGN Protocols
- 11** Automated Fuzzing
- 12** Fuzzing NGN Networks
- 13** Benefits Of Proactive Testing

CODENOMICON Ltd. | info@codenomicon.com | www.codenomicon.com

Tutkijantie 4E | FIN-90590 OULU | FINLAND | +358 424 7431
12930 Saratoga Avenue, Suite B-1 | Saratoga, CA 95070 | UNITED STATES | +1 408-414-7650

PREEMPTIVE SECURITY AND ROBUSTNESS TESTING SOLUTIONS

Abstract— Telecom networks are increasingly synonymous with the Internet and mobile phones resemble computers in all aspects, except the level of protection. By fuzzing protocol implementations before deployment, critical vulnerabilities can be found and fixed making networks and devices more robust against attacks.

Keywords- Security; Next Generation Networks; Protocol Testing; Fuzzing; Vulnerability

1 Introduction

Telecommunication networks have not encountered any large scale security attacks. GSM and various descendant technologies do contain security flaws, which can be exploited for fraudulent purposes. An example, of such an inherent weakness is unilateral authentication. This means that the network authenticates the user, but the user does not authenticate the network. Therefore, an attacker with a false base station server can impersonate a valid operator charging customers and operator's high prices for traffic. In the design phase of GSM, such attacks were deemed impractical, due to high price of the equipment involved, but equipment prices have since dropped making such attacks completely applicable. In our own test labs, we have been able to verify the possibility of such attacks with freely available equipment [1].

However, the lack of suitable tools and motivation has protected telecommunication networks and handsets. With the arrival of Smartphones, all-IP Next Generation Networks (NGNs) and new more powerful access technologies, such as eNodeBs, all this is changing [2]. Mobile phones are now used to handle more valuable data, increasing the motivation to hack them [3]. Telecom networks are increasingly synonymous with the Internet and mobile phones resemble computers in all aspects, except the level of protection. It is only a matter of time, before Telecom networks start encountering their share of security incidents [3]. In this paper, we discuss security challenges in NGNs and look at how proactive protocol testing methods can be used to make NGNs more secure.

2 Vulnerabilities Enable Attacks

There are numerous different types of malware, including viruses, worms, Trojans, backdoors, keystroke loggers, rootkits and spyware. What all these attacks have in common is that the initial access is enabled by a vulnerability in the code. Vulnerabilities are unpatched software flaws. Software contains flaws either in the form of outright implementation mistakes or subtler design issues. If these flaws are not fixed before deployment, they become vulnerabilities.

Hackers need to find a vulnerability in the protocol implementations in order to devise an attack against the target system [3]. Once they have gained initial access, they can instruct the system to download more software giving them more control over the infected device [3]. If a single device within a network becomes compromised, for example a mobile phone in a corporate network, it puts all other devices in the network under risk [3]. By creating networks of slave mobile devices, known as botnets, hackers can carry out attacks against Telecom networks on a previously inconceivable scale. Combined with the fact that, unlike traditional telecom networks, all-IP networks can create multiple simultaneous connections, this creates a very real risk of Denial of Service (DoS) attacks in Telecom networks [4].

3 Reactive Defenses Are Not Enough

As software, malware is easy alter and to update [3]. Thus, it is hard to detect and once a system has been compromised, it is almost impossible to remove the malicious code completely [3]. Most security technologies, such as intrusion detection systems (IDS) products, are reactive meaning that they try to stop attacks instead of preventing them proactively. Moreover, they are frequently signature based meaning they can only detect pieces of malware, for which an identifier, known as a signature, already exists and has been deployed. Thus, they can only find vulnerabilities that have already been reported. This is especially problematic with new technologies, which are typically prone to vulnerabilities, and proprietary protocol extensions, for which there are no ready bug lists, and thus it is impossible to find weaknesses using vulnerability or security scanners [5].

4 Discovering Vulnerabilities with Fuzzing

The best way to avoid attacks is to get rid of exploitable vulnerabilities proactively. Vulnerabilities are introduced into software during design and implementation. It is not always easy

The complexity of the new technologies also increases the probability of vulnerabilities. The more complex a software is, the more likely it is to contain implantation mistakes [10]. For example, with Triple play and IP Multimedia Subsystem (IMS), the various interfaces, players, protocols and applications are source of such complexity that it already has significant security implications [11]. With such complex technologies vulnerability management is also necessary from an interoperability point-of-view [11].

Vulnerabilities affect the Quality of Service (QoS), this is particularly challenging for new IP technologies, like VoIP and Internet Protocol TV (IPTV), which are replacing traditional voice and broadcasting services. Users are used to the high quality transmission of the traditional telecom networks and television broadcasts, thus both VoIP calls and television programs transmitted over an IP network are highly sensitive to packet loss and jitter [12].

6 Fuzzing in the ITU-T X.805 Security Architecture

The ITU-T X.805 standard defines the basic ITU-T security architecture for NGN networks. It consists of three layers and three planes. [13]

The layers are a hierarchical grouping of network functions and elements. The bottom layer is the infrastructure security layer, which consists of the physical building blocks of a NGN. The second layer is the services security layer, which contains the services provided to the end-users, such as VoIP, WiFi or Location Services. The third layer contains the network-based applications, which build upon the services. These applications include e-commerce, web browsing and email. [13]

The security planes represent different types of network activity. The end-user security plane encompasses the access and use of the network by customers. The control/signaling plane consists of activities which enable the efficient functioning of the network. The management security plane is for the management and controlling of network elements, services and applications. [13]

Each security plane is applied to each security layer creating nine security perspectives, each with its own vulnerabilities and threats. In each security perspective there are eight security perspectives, which need to be considered, namely [13]:

- Access control
- Authentication
- Non-repudiation

- Data confidentiality
- Communication security
- Data integrity
- Availability
- Privacy

The purpose of the security model is to make it easier to determine the necessary security countermeasures. Even though no actual tools or methods are mentioned, the security dimensions clearly focus on protecting systems, which are already in operation. Proactive protocol testing, i.e., fuzzing, can be used to find security issues, before they become acute [8].

Fuzzing can be used to test the entire protocol stack, and thus it supports the layer thinking of the ITU-T X.805 model. However, the security plane thinking is redundant for protocol testing, because it is intended for protecting a system in operation, whereas protocol testing focuses on testing systems, before deployment and integration. Fuzzing can also be used to test systems in operation, but not in a live environment, because the tests trigger vulnerabilities with simulated attacks [14].

7 ITU-T Standard Y.2700

The ITU-T standard Y.2700 on NGN network security defines three types of security zones: trusted, trusted but vulnerable and un-trusted [13].

In the trusted zone, the network elements and system are fully controlled by the NGN provider and they only communicate with trusted and trusted but vulnerable network elements. However, it should not be taken for granted that the trusted network elements are secure per se. [13]

Elements and systems in the trusted but vulnerable zone communicate with both trusted and un-trusted elements. Located on the border between the trusted and the un-trusted zone, these network elements have an important role in protecting the trusted network zone from network attacks. However, if they are vulnerable, they merely convey the attack to the trusted zone. [13]

In the un-trusted zone, the NGN provider has no guaranteed control over the network equipment. Therefore, it may be impossible to force their security policies on these elements. [13] The same type of security approach based on the openness of interfaces is also applied to fuzzing. In fuzzing, testing focuses

on open network interfaces which can be accessed from the outside meaning the trusted but vulnerable network elements [8].

8 NGN Network Interfaces

The NGN architecture (Y.2012) supports two reference points to end user functions, namely the user-to-network interface (UNI) and the network-to-network interface (NNI), and a reference point to the functional group of applications, namely the application-to-network interface (ANI). The UNI connects the NGN network to both NGN and legacy terminal equipment and customer networks. The NNI connects the NGN network to other NGN providers' networks, IP multimedia networks and PSTN/ISDN networks. The application functions group consists of applications such as e-commerce, web browsing and email. [13]

Sometimes the DUT setup is not so straightforward. We have observed some difficulties, especially with in-car Bluetooth systems. These systems might check (using SDP) the supported profiles from tester as a part of pairing procedure and refuse to establish the trust relationship if the tester does not respond to the SDP query with right profile information. Sometimes it can be quite difficult to know exactly what the DUT is expecting. We have equipped our tester with SDP server which contains profile information which should be accepted by most devices.

Some (especially in-car) systems may actually contain two distinct Bluetooth stack implementations. One for phone call related profiles and one for audio playback. In this case the tester needs to be paired with both stack instances separately. Also, don't forget to test the core protocols on both stacks since the stacks may very well be completely different from each other.

9 Critical Interfaces in NGN

The most critical interfaces in all-IP NGN networks are the interface between user access plane and the core network (UNI) and the interface between the core network and the Internet. These are the two interfaces that carriers can least control. The problem in the off-site connections is not the wireless connection in itself, but potential vulnerabilities in the higher level protocol implementations in the border elements. [15]

Particularly, vulnerabilities in IPv4 and IPv6 implementations are dangerous, because they can be used to directly access components in the operator's core network. As an entry vector,

the IP layer is visible to most users [15]. It is also the easiest one to attack, due to the large number of ready tools used for Internet hacking. Additional security, quality and interoperability challenges are created by the transition from the matured IPv4 to the emerging standard IPv6 [11].

UNI is greater attack vector in NGN than traditional telecommunication networks, due to the introduction of more powerful access technologies. Traditionally, base stations have had limited functionality, but with modern access point base stations, such as FemtoCells and Home eNodeBs, some logic, like transmission, are handled on new access point base stations. This will achieve shorter response times, but, at the same time, it enables easier outside access into the carrier's core network. New types of home routers also allow users to program the handset radio features. [15]

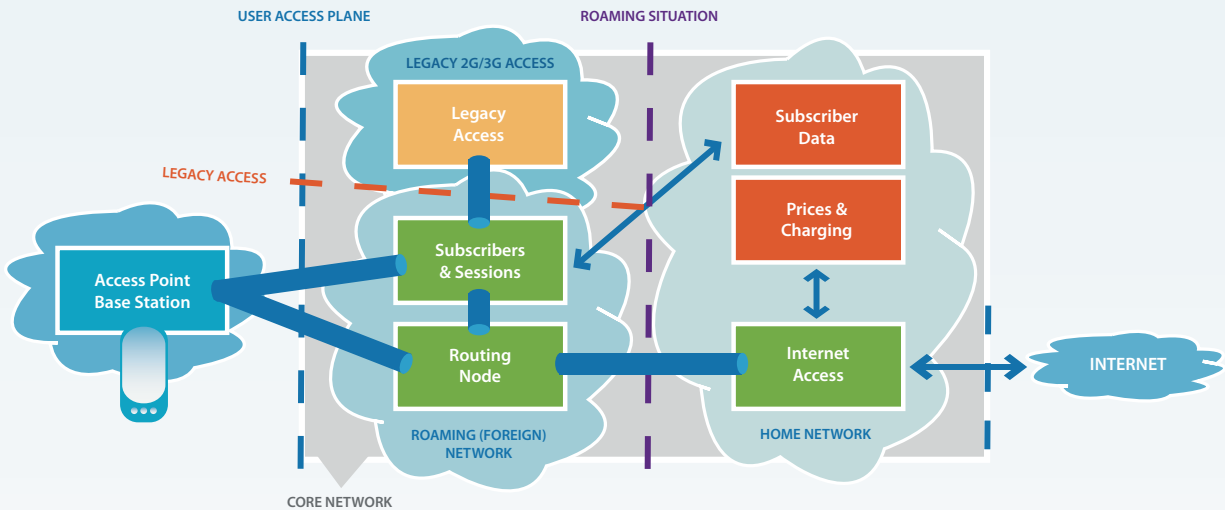
Other vulnerable interfaces are created by legacy access and roaming scenarios. Roaming situations create a trust boundary between two carriers within their core networks and faulty legacy equipment, for example, in the roaming partner's network, can send invalid data, which trigger vulnerabilities in the operators own network. See Figure 2 for a depiction of vulnerable interfaces in operator networks. [15], [11]

10 NGN Protocols

The NGN architecture consists of two-layered IP interfaces with one layer tunneled inside the other, see Figure 2. The GPRS Tunneling Protocol (GTP) and Proxy Mobile IP tunnel IP-based user plane data from client terminals such as mobile phones, thus enabling users to move from one place to another while continuing to connect to the internet to the internet as if from one place. GTP also ensures the transfer of data between networks, which would otherwise be incompatible. Within the core network, Diameter is used to transfer subscriber data. Diameter is an Authorization, Authentication and Accounting (AAA) protocol, which uses TCP or SCTP for transportation. In current 3G networks, the IP-based Diameter is already used for network signaling in order to implement authentication, policy control and charging. [2]

11 Automated Fuzzing

In fuzzing, thousands and even millions of misuse-cases are created for each use-case, thus most robustness testing solu-



tion contain at least some degree of automation. There are two popular ways to automate fuzzing: mutation-based and model-based fuzzing. [16]

Mutation-based fuzzing uses samples of real-life inputs, like network traffic and files, as basis for testing. The outgoing test cases can be generated by modifying the samples randomly or based on the sample structure. The structure can either be defined manually by providing the fuzzer with metadata. Some fuzzers can also, to some extent, automatically deduce the structure from message samples.

In model-based fuzzing, the process of data element identification is already automated by using specifications, which also provide the model with other protocol or file format specific information, e.g., on the boundary limits of the data elements. Sample-based mutation is also highly dependent on the quality of samples used: Elements that are not included in the samples cannot be properly tested without additional model. Thus, model-based test generation is often more systematic and has higher coverage than mutation based test generation. [16]

Model-based test case generation also significantly reduces the number of test cases needed, because the protocol and file format specific information from the specifications can be used to target the test cases. Being based on specifications, model-based fuzzers contain all the protocol messages, thus they can genuinely interoperate with the tested devices or system, thus they can find more hard-to-reach vulnerabilities and test interoperability. [16]

12 Fuzzing NGN Networks

NGN is an emerging standard, and thus the NGN protocols are fairly well specified. Clear specifications already exist for the core NGN protocols, such as GTP, Diameter and Proxy Mobile and the underlying layers consist of standard protocols, like IP, Mobile IP, TCP and SCTP. Thus, both the core protocols and the underlying protocols can be tested with intelligent model-based fuzz tests. In addition, the IP protocol can be used to test end-to-end connectivity within the network, because all communication within NGN networks is IP-based. Testing the IP-layer is important, because not only is it the most open protocol being used it is also the easiest one to attack, due to the large number of ready tools. [2]

Fuzzing can also be used to test the services layer. Technologies like VoIP, Wi-Fi and IPTV also consist of layers of protocols making it possible to test the entire infrastructure, see Table 1. For example, fuzz testing can be used to test the complete IPTV infrastructure, including IPTV Content Source systems, Delivery and Management networks and Home Networks. With VoIP services, protocols like SIP and RTP are becoming an increasingly important part of telecommunication infrastructure. In the wireless world, anything can be attacked, anyone can attack and an attacker can remain anonymous. Yet, the adoption rate of Wi-Fi has skyrocketed. Thus, the security and robustness of Wi-Fi Access Points and Wi-Fi enabled client devices needs to be ensured.

NGN networks are backwards compatible supporting legacy 2G and 3G access technologies. These legacy technologies contain rarely used features, like WAP, which are not subjected to rigorous everyday use and can thus cause multiple problems in the network. [2]

13 Benefits of Proactive Testing

The ITU-T standards for securing NGN networks focus on networks, which are already in operation. Fuzzing introduces the possibility of proactive security. Instead of waiting for others to discover security issues, NGN providers can actively reveal vulnerabilities using fuzzing tools. By discovering issues proactively they gain more time to fix them and can also avoid problems. The use of fuzzing is not restricted to in-house software developments. Fuzzing third-party software and equipment before integration can help avoid costly security, quality and interoperability problems. A major US operator uses fuzz tests as acceptance criteria for selecting vendors. In their tests, they have found significant differences in security and robustness between different vendor equipment.

By using Fuzzing as a part of their security routine, operators can improve their Quality of Service, namely availability, ensure interoperability and improve communication security within their networks.

NGN	VoIP	Wi-Fi	IPTV
SCTP	SCTP	802.11 WPA	IPv4
GRE	H.248		IPv6
IPSec	H.323		TLS/SSL
Diameter	RTSP		IPSec
LDAP	TLS/SSL		RTP/RTCP/ SRTP
TLS/SSL	SIP		RTSP
SIP	SigComp		HTTP
GTP	RTP		TFTP
RADIUS	RTCP		FTP
PMIP	SRTP		PIM-SM/DM
IPv4	MGCP		RSVP
IPv6	UPnP		IGMP
			MPEG4

REFERENCES

- [1] M. Varpiola (personal communication, June 2011).
- [2] S. Petäjäsöja & J. Lämsä (personal communication, March 2010).
- [3] OECD, "Malicious Software (Malware): A Security Threat to the Internet Economy." OECD Ministerial Meeting on the Future of the Internet Economy, June 2008.
- [4] Nokia Siemens Networks, "Adapting commercial Next Generation Network Solutions to satisfy military and other special real-time requirements".
- [5] A. Takanen, "Fuzzing: Helping to Avoid Zero-Day Attack", February 2010.
- [6] D. Gollman, "Computer Security", Wiley & Sons, 1999.
- [7] B.P. Miller & al., "Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services", University of Wisconsin, April 1995.
- [8] A. Takanen, J.D. Demott & C. Miller, Fuzzing for Software Security Testing and Quality Assurance, Artech House, 2008.
- [9] C. Cowan, "StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks". The USENIX Association, "Proceedings of the Seventh USENIX Security Symposium", January 1998.
- [10] B. Schneier, "Software Complexity and Security", Cryptogram, 2000.
- [11] A-M. Juuso & A. Takanen, "Unknown Vulnerability Management for Telecommunications, March 2011.
- [12] Nokia Siemens Networks, "High-quality and resilient IPTV multicast architecture".
- [13] Telecommunication Standardization Sector of ITU, "ITU-T Recommendation Y.2701: Security Requirements for NGN release 1", "Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks", April 2007.
- [14] A-M. Juuso & A. Takanen, "Building secure Software Using Fuzzing and Static Code Analysis", November 2010.
- [15] S. Petäjäsöja, J. Lämsä & A-M. Juuso, "Hot Fuzz", Professional Tester, March 2010.
- [16] R. Kaksonen & A. Takanen, "XML Fuzzing Tool: Testing XML on Multiple Levels", Testing Experience, December 2009.

Paper accepted for presentation at the "Technical Symposium at ITU Telecom World 2011"
<http://world2011.itu.int/>

CODENOMICON Ltd.
info@codenomicon.com
www.codenomicon.com

Tutkijantie 4E
FIN-90590 OULU
FINLAND
+358 424 7431

12930 Saratoga Avenue, Suite B-1
Saratoga, CA 95070
UNITED STATES
+1 408 252 4000

25/F., Queen's Road Centre
152 Queen's Road Central
HONG KONG
+852 3426 22900

21 Science Park Road
#02-01 The Aquarius
SINGAPORE 117628
+65 9186 8174