SEVENTH FRAMEWORK PROGRAMME

Information & Communication Technologies
Secure, dependable and trusted Infrastructures

COORDINATION ACTION
Grant Agreement no. 216331

forward

Managing Emerging Threats in ICT Infrastructures

Deliverable D3.1: White book: Emerging ICT threats

Contractual Date of Delivery	31/12/2009
Actual Date of Delivery	17/01/2010
Deliverable Security Class	Public
Editor	FORWARD Consortium
Contributors	FORWARD Consortium
Quality Control	FORWARD Consortium

The FORWARD Consortium consists of:

Technical University of Vienna	Coordinator	Austria
Institut Eurécom	Principal Contractor	France
Vrije Universiteit Amsterdam	Principal Contractor	The Netherlands
ICS/FORTH	Principal Contractor	Greece
IPP/BAS	Principal Contractor	Bulgaria
Chalmers University	Principal Contractor	Sweden

Contents

1 Executive Summary and Main Recommendations 7

2 Introduction: Threat List 11

3 Threat Category: Networking 15
3.1 Overview 15
3.2 Routing infrastructure 17
3.3 IPv6 and direct reachability of hosts 18
3.4 Naming (DNS) and registrars 20
3.5 Wireless communication 22
3.6 Denial of service 24

4 Threat Category: Hardware and Virtualization 27
4.1 Overview 27
4.2 Malicious hardware 27
4.3 Virtualization and cloud computing 29

5 Threat Category: Weak Devices 31
5.1 Overview 31
5.2 Sensors and RFID 32
5.3 Mobile device malware 34

6 Threat Category: Complexity 39
6.1 Overview 39
6.2 Unforeseen cascading effects 39
6.3 Threats due to scale 41
6.4 System maintainability and verifiability 43
6.5 Hidden functionality 44
6.6 Threats due to parallelism 45

7 Threat Category: Data Manipulation 47
7.1 Overview 47
7.2 Privacy and ubiquitous sensors 47
7.3 False sensor data 48

7.4 Threats related to social networks 50
7.5 Online games 52

8 Threat Category: Attack Infrastructure 53
8.1 Overview 53
8.2 Underground economy support structures 54
8.3 Advanced malware 55

9 Threat Category: Human Factors 59
9.1 Overview 59
9.2 User interface 60
9.3 The insider threat 61
9.4 Safety takes priority over security 62
9.5 New vectors to reach victims 63
9.6 Targeted attacks, spear phishing 65

10 Threat Category: Insufficient Security Requirements 67
10.1 Overview 67
10.2 Retrofitting security to legacy systems 68
10.3 Use of COTS components 69
10.4 Next generation networks 70

11 Introduction: Scenarios 75

12 Scenario: Election Fraud 77
12.1 Overview 77
12.2 Scenario 77
12.3 Consequences 80
12.4 Related threats 80
12.5 Could it happen? 81

13 Scenario: Crashing the Stock Market for Fun and Profit 83
13.1 Overview 83
13.2 Scenario 83
13.3 Consequences 85
13.4 Related threats 86
13.5 Could it happen? 86

14 Scenario: Industrial Espionage 87
14.1 Overview 87
14.2 Scenario 87
14.3 Consequences 90
14.4 Related threats 90
14.5 Could it happen? 91

15 Scenario: The Smart Grid 93
15.1 Overview 93
15.2 Scenario 93
15.3 Consequences 95
15.4 Related threats 96
15.5 Could it happen? 96

16 Scenario: The Oil Spill 99
16.1 Overview 99
16.2 Scenario 99
16.3 Consequences 102
16.4 Related threats 102
16.5 Could it happen? 103

17 Scenario: Fabrication Take-over 105
17.1 Overview 105
17.2 Scenario 105
17.3 Consequences 106
17.4 Related threats 107
17.5 Could it happen? 107

18 Scenario: The Politician's Phone 109
18.1 Overview 109
18.2 Scenario 109
18.3 Consequences 112
18.4 Related threats 112
18.5 Could it happen? 113

19 Scenario: Mass Blackmailing through Social Networks 115
19.1 Overview 115
19.2 Scenario 115
19.3 Consequences 117
19.4 Related threats 117
19.5 Could it happen? 117

20 Scenario: Attack against an Electric Power Station 119
20.1 Overview 119
20.2 Scenario 119
20.3 Consequences 121
20.4 Related threats 121
20.5 Could it happen? 122

21 Scenario: Attack on a Water Power Station 123

21.1 Overview 123

21.2 Scenario 123

21.3 Consequences 125

21.4 Related threats 125

21.5 Could it happen? 125

Chapter 1

Executive Summary and Main Recommendations

Predicting the future is notoriously difficult. Nevertheless, one main goal of the FORWARD project was to identify future attacks that threaten the security of the ICT infrastructure of Europe. It is clear that ICT infrastructures are no longer independent of societies and the economy, but they have evolved into the enabling and carrying platform that facilitates communication, production, trade, and transportation. Thus, attacks against ICT infrastructures have significant, adverse impact on the well-being of Europe and its people, and it is clear that adequate protection must be provided. By identifying future problems, Europe can take an active approach in defending against anticipated threats. In particular, one can prepare by setting a research agenda that provides solutions before threats become unmanageable.

Over the last two years, members of the project consortium have worked with international experts from academia and industry (both from Europe and the rest of the world) to identify future threats, finding those that are most pressing and that require the most immediate attention. To achieve this, three working groups were formed: One group focused on threats related to malware and fraud, one group studied the emerging smart environments, and one group looked at critical systems. In addition, we held two workshops that were attended by almost two hundred people, who gathered in Goteborg and Nice and discussed potential future threats. To explore emerging and future threats in a systematic fashion, we defined four axes along which future developments are anticipated or are currently unfolding. These axes, which are *new technologies*, *new applications*, *new business models*, and *new social dynamics*, served as the framework to guide our analysis.

As a result of the work in this project, 28 threats were identified that we classified in the following eight categories:

1. *Networking:* This category covers threats that are related to the introduction and deployment of new (often wireless) network technologies, but it also covers emerging threats against infrastructure services (routing, DNS) on the current Internet.

2. *Hardware and virtualization:* This category covers threats due to new hardware and software developments that allow computation to be moved to virtual computers, and ultimately, the cloud. It also covers malicious hardware.

3. *Weak devices:* This category covers threats that are introduced with new computing devices that are limited, both computationally and because of power constraints. The problem is that security is "expensive," and weak devices might not be able to afford to implement and run adequate protection mechanisms.

4. *Complexity:* This category covers threats that emerge due to the fact that some future systems will contain billions of components. Another source of complexity are large monolithic systems that offer more and more functionality. The increased complexity leads to unexpected and unintended dependencies, interactions, and security consequences.

5. *Data Manipulation:* This category covers threats that stem from the fact that people (and systems) store more data online, and this data is becoming increasingly valuable and sensitive.

6. *Attack infrastructures:* This category covers threats that are related to the fact that adversaries actively develop and deploy offensive platforms (such as botnets). That is, adversaries no longer perform hit-and-run attacks, but they establish operational bases on the Internet used to carry out malicious campaigns.

7. *Human factors:* Human factors always played a role in security. This category covers threats that are due to increasing concerns over insider attacks, especially in the context of outsourcing. The category also covers threats that are related to new social engineering attacks.

8. *Insufficient security requirements:* This category covers problems and threats related to legacy and commercial-off-the-shelf systems that have not been built with sufficient protection and are now used and deployed in scenarios for which their protection mechanisms are inadequate.

For all 28 threats that were identified, the consortium, together with experts and researchers from academia and industry, attempted to quantify the urgency of additional (research) efforts that would be needed to mitigate each threat. This "ranking" process was based on four factors: The severity of a threat, the expected likelihood that it would become prevalent, the lack of awareness in the community about a threat, and the consideration of existing efforts that are already underway. Based on our analysis, we found the following five threats to be the ones that deserve the most urgent attention (Top-5 threats):

- *Threats related to parallelism:* Single processors have hit the CPU speed wall. However, Moore's law continues to hold, and processor manufacturers are now shipping machines with many CPU cores. These multi-cores need to be programmed, and the paradigm shift from sequential to parallel programming will likely bring a wide range of new vulnerability classes that we need to mitigate. Thus, we require new techniques to help developers write correct code and to detect bugs in parallel programs.

- *Threats related to scale:* The effects of scale can be felt everywhere on the Internet. This ranges from the sheer number of devices connected to the network to the size and complexity of individual software packages. We need ways to manage the complexity, scale, and security of such systems.
- *Underground economy support structures:* Many attacks on the Internet are driven and fueled by a thriving underground economy. This is the result of a paradigm shift from "hacking for fun" to "hacking for profit." Unfortunately, the mechanics of the underground economy and its support structures are poorly understood. However, it is necessary to study and actively combat the root cause that drives such diverse threats as botnets, phishing, and spam.
- *Mobile device malware:* Malware is already a significant problem on today's Internet. Consider that the number of mobile devices is growing rapidly, users get more comfortable downloading and installing applications (e.g., via Apple's AppStore), and phones are increasingly used for critical applications (e.g., for online banking). Thus, it is just a matter of time before mobile device malware will become mainstream. Unfortunately, mobile devices are constrained, both computationally and because of power limitations, making it hard to deploy costly, traditional anti-malware techniques. As a result, better malware defenses are crucially required for mobile devices.
- *Threats related to social networks:* Social networks are regularly used by hundreds of millions of users who provide a wealth of private information online that could be abused. In addition, social network providers have been notoriously unwilling to provide sufficient privacy protection for their users, and they are looking for ways to monetize their audience and the data they upload. This is a dangerous combination that provides attackers with new ways to reach (and scam) victims, and it can lead to severe, large-scale data theft.

This report is divided into two main parts:

The *first part (Chapters 2 to 10)* provides the complete list (and ranking) of the emerging and future threats. It gives a more detailed discussion for each threat and provides an overview that shows how each threat is rated according to our factors that were used to determine a threat's urgency.

The *second part (Chapters 11 to 21)* discuss ten threat scenarios. These scenarios show how adversaries can leverage the previously-identified threats to realize their malicious goals. We believe that such a collection of scenarios, which describes the real-world impact of attacks, is a valuable complement to the raw list of threats. The scenarios can illustrate future dangers, and we hope that they provide inspiration to support research in security-critical areas.

Recommendation 1: The EC should stimulate efforts that carry out research and development of techniques and systems (tools) for protecting against emerging ICT threats. The priority areas are:

- Protection of systems that are difficult to build, manage, and understand due to their scale and complexity
- Protection against malicious code (malware)
- Protection against threats that compromise users' privacy, particularly those on online social networks

Recommendation 2: The EC should support ongoing efforts to monitor developments in the ICT threat landscape. The threat landscape often changes rapidly and unpredictably as new technologies are deployed or new attacks are discovered. One requires an established and prepared entity to quickly react to these changes and assess the threat potential of new developments.

Recommendation 3: The EC should support awareness initiatives and programs to educate its citizens about online threats and possible preventive actions. The reason is that certain threats cannot be addressed by technical means alone. Instead, defenses rely on proper reactions from informed users.

Recommendation 4: The EC must recognize that ICT infrastructure is interconnected and deployed on a global scale. Hence, particular emphasis must be put on international collaborations and initiatives.

Recommendation 5: The EC should (continue to) encourage interdisciplinary work and initiatives to bring together researchers from academia and industry as well as policymakers to cooperate on finding solutions to the threats against ICT infrastructure.

Chapter 2

Introduction: Threat List

The first part of this document describes the identified threats in more detail. This list of threats was compiled in two phases:

For the first phase, the three working groups of the Forward project compiled three individual lists of threats that related to their respective topics and domains. These individual lists, as well as the process in which these lists were derived, was explained in a previous deliverable D2.1.x: “Individual Threat Reports.” As mentioned in that document, the threats and the trends were independently identified by each working group. As a result, some threats are similar. For example, the Malware and Fraud working group has listed mobile malware as an emerging threat. Similarly, the Smart Environments working group is also discussing this threat in its report. As some of the threats are multi-faceted, it is not surprising that there is a certain overlap between some of the threats.

For the second phase, the task was to identify overlapping threats and to condense them into a coherent and unified list. Moreover, the individual reports did not yet contain a risk assessment for individual threats. This risk assessment, that is, our view on the severity and likelihood of each threat, was also part of the compilation of the threat list that is described in the following chapters.

The approach to derive the unified threat list was a two-step process that was finalized during a one day, in-person, executive board meeting held in Vienna in November 2009. First, the partners carefully went through the individual lists and looked for threat titles that appeared to be closely related. Here, the advantage of the in-person meeting became evident, as different working groups sometimes had quite different views about the actual meaning of a particular title for a certain threat. Once we had derived a unified list, we also decided to group related threats into eight different topical categories, as presented in the following chapters.

The second step was based on the unified threat list, and it consisted of a ranking of these threats. We identified a total of 28 threats, and while it is useful in some cases to have at one’s disposal a comprehensive view over the expected threat landscape, there are other situations in which one has to set priorities and address the most pressing threats first. To this end, we require some kind of way to compare threats with regards to the urgency in which we think they need to be addressed. Of course, this is a difficult and often subjective task. Thus, we felt that it might not be possible to provide an absolute ranking that assigns a specific and final position from 1 to 28 to each threat. However, we agree that some kind of ordering is necessary. As a result, we decided that it would be best to group threats into three broader classes that correspond to a *low*, *medium*, and *high* priority. Within each class, threats can be considered equal. This ranking also provides policymakers with flexibility to prioritize threats within each class according to their constraints and views. Also, note that the ranking (classes) proposed in this step is independent from the eight topical categories used to group threats.

To determine the priority class for each threat, we used the following four factors that capture different, important aspects of a threat:

- **Impact:** This factor describes how many users are affected and what damage level is to be expected. Clearly, the severity and impact of a threat is important to determine the priority. When many users are affected from a threat and suffer significantly from it, then it is more important to address and mitigate it.

- **Likelihood:** This factor captures the (expected) probability that a threat in question is actually carried out. This captures the difficulty to mount a certain attack, and the probability that an adversary gets caught. Also, this factor reflects the uncertainty in our belief that a threat will become a reality. For example, when a certain threat requires technical advances that are not expected before the next five years, then the likelihood is lower than for an already emerging threat that can be observed in the wild.

- **Obliviousness:** This factor captures the lack of awareness of the public and the research community for a threat. The higher the obliviousness, the more likely it is that a threat will not be addressed in a timely fashion before it is realised. As a result, there is the need to generate more awareness about a threat.

- **R&D Needs:** This factor captures the extent to which new research and development efforts are needed to mitigate a threat. In some cases, effective mitigation efforts are known, but they would require proper policy or investment to put into action. In this case, the R&D needs would be considered low.

For each threat, we evaluated all four factors and assigned one of the three classes low (L), medium (M), or high (H), respectively. Then, we evaluated each threat holistically (based on all factors) and determined the final priority class.

In the following, we first provide the entire list of threats, grouped by the three priority classes. Then, we describe and discuss every threat into more detail. For this discussion, we used the eight topical categories.

High Priority					
#	Threat Description	Impact	Likely	Oblivious	R&D
1	Threats due to parallelism	M	M	H	M
2	Threats due to scale	H	M	H	M
3	Underground economy support structures	H	H	L	H
4	Mobile device malware	H	H	M	H
5	Threats related to social networks	H	H	M	H

Medium Priority					
#	Threat Description	Impact	Likely	Oblivious	R&D
6	Routing infrastructure	H	H	L	M
7	Denial of service	H	H	L	M
8	Wireless communication	H	H	M	M
9	Unforeseen cascading effects	H	M	H	H
10	False sensor data	H	M	H	M
11	Privacy and ubiquitous sensors	M	M	M	M
12	User interface	M	H	M	H
13	The insider threat	H	M	M	M
14	System maintainability and verifiability	M	H	M	M
15	Hidden functionality	M	M	H	M
16	New vectors to reach victims	M	H	M	H
17	Sensors and RFID	M	H	M	H
18	Advanced malware	M	H	M	M
19	Virtualization and cloud computing	H	M	H	M
20	Retrofitting security to legacy systems	M	M	M	L
21	Next generation networks	H	H	M	M

Low Priority					
#	Threat Description	Impact	Likely	Oblivious	R&D
22	IPv6 and direct reachability of hosts	M	H	M	M
23	Naming (DNS) and registrars	L	H	M	L
24	Online games	L	H	M	L
25	Safety takes priority over security	L	M	H	M
26	Targeted attacks	M	H	M	M
27	Malicious hardware	M	L	H	M
28	Use of COTS components	M	H	M	M

Table 2.1: List of future and emerging threats.

Chapter 3

Threat Category: Networking

3.1 Overview

The Internet is a communication environment that has become an essential part of our everyday life, in the same fashion as the electricity or the telephone network have become essential over the last one hundred years. The more products and services we access through it, the more dependent we become on its functionality and availability. Indeed, the "functionality" of the Internet has outgrown its initial goal, to transfer information between distant sites; we now expect it to transfer trust and to operate in new critical areas.

Many critical applications highly depend on its existence and that Internet-based services can be offered with a desired level of availability and security. This is not easy, however, since the protocols used are quite old and were designed long before the Internet was even being thought of. Moreover, there are many forces that try to disrupt or interfere with these services. There is a steady noise of unwanted traffic on the Internet and systems are constantly probed for weaknesses, where the vulnerable systems are then inevitably exploited. Non-intentional threats also exist that may affect the services the Internet can offer, for example, improperly designed applications that are not robust or reliable enough to support their intended use or the problems caused when the devices supporting the network infrastructure are misconfigured and cease to function.

The main threats can be divided into a few general categories:

- *Attacks against the infrastructure of the Internet,* such as against routers and routing algorithms.

- *Denial of service attacks* where strategic links or essential backbone nodes are taken out of service.

- *Wire-tapping attacks* where the confidentiality or integrity of traffic is compromised, both on wired and wireless links.

End-systems connected to the Internet are also possible targets for attacks with the goal to disrupt their services.

- Improper design or improper use of the services that the Internet offers, for example, the design of mission-critical systems that are accessible from the Internet and possibly in turn also depend on its services.

- Denial of service attacks against servers on the Internet, for example, by exploiting known vulnerabilities in applications or systems.

- Distributed denial of service attacks, where the Internet infrastructure and the large number of unprotected nodes on the Internet are used to drown selected sites in traffic.

Most of the attacks rely on the existence of vulnerabilities in networking equipment, clients, servers, and protocols but also on the improper configuration of systems. For example, nowadays most Internet providers ship to their customers out-of-the-box (self-configured) network devices such as ADSL and wireless routers. These devices are quite easy to use and install. Often, they come with default settings. Unfortunately, because of their ease of use, these devices may sometimes have default configurations that may be insecure. They may have default authentication credentials, weak keys, and may allow open Internet access to outsiders. Many of the attacks on the Internet today target personal computers as many home computers have high-Internet connectivity that can be useful for the miscreants (e.g., for launching DoS attacks or for sending spam). Hence, whereas many attacks were targeting servers 10 years ago, we see a strategic shift by the attackers who are less interested in servers that have become more difficult to attack (e.g., because of default firewalls and automatically installed patches), but more interested in home computers.

The size of the Internet and the extremely large number of systems connected to it make it impossible to aim for complete security and reliability (as further elaborated in Section 6). However, a sound infrastructure should be present to make it possible to offer secure and reliable services over the Internet. And with a reasonable effort, it should be possible to set up end systems, such as servers and clients, in such a way that their services can be trusted. In order to do this, it is necessary to know what shortcomings we face today, what the fundamental laws of the Internet are, and possibly also considering some changes in the infrastructure in order to reach the intended service, reliability and security we need in the future.

Research is required on attacks that target the computing infrastructure of home users as well as the Internet backbone. Interestingly, it was known for a long time that BGP is vulnerable to attacks. However, the problem was ignored by the industry. Many experts believe that the number of attacks against the Internet infrastructure will increase in the future.

3.2 Routing infrastructure

Threats. To date, there have not yet been major attacks against routers that use the Border Gateway Protocol (BGP), the main routing protocol that regulates the way that routes are established between autonomous systems (ASes) on the Internet. That is, they have at least not been documented publicly. As a result, the security community has not paid sufficient attention to studying the threats against the Internet routing infrastructure.

In practice, routing misconfigurations are common. A misconfiguration can cause effects that are similar to a real attack. In fact, a misconfiguration can result in serious reachability problems that are analogous to denial of service attacks. For example, in April 1997, the AS7007 incident was caused by a misconfigured router that flooded the Internet with incorrect advertisements, announcing AS7007 as the origin of the best path to essentially the entire Internet [115].

In a typical scenario, one can assume that the attacker has managed to compromise one or more BGP routers in the Internet infrastructure. Note that such an assumption is reasonable as it has been shown in the past that many routers and networking devices on the Internet are not well-protected [156]. In such an attack, the attacker could aim to perform traffic redirection, sniffing, creating a routing instability, or traffic subversion.

A compromised router can typically modify, drop, or introduce fake BGP updates. As a result, other routers could have an incorrect view of the network, which could lead to blackholing, redirection, or instability.

As discussed in [115], the effectiveness of some attacks depends on the AS topology and on the location of the compromised router relative to the victim network. False updates and prefix hijacking are probably the most straightforward type of BGP attack. They occur when an AS announces a route that it does not have, or when an AS originates a prefix that it does not own.

Another possible attack is route de-aggregation. When used as an attack, it breaks up an address block into a number of more specific prefixes. Since the BGP route selection process gives higher preference to the longest matching prefix for a given destination, the attacker can use de-aggregation to announce fake routes that will be preferred throughout the Internet over the legitimate routes to that network.

Instability, in the form of wide-scale cascading failures, can occur when a number of BGP sessions repeatedly time-out due to router reboots, link congestion, or physical link intermittent failures. Instability, in the form of delayed convergence, can also occur upon routing or policy changes, due to the way BGP explores alternate paths.

We expect to see more routing-based attacks in the future that involve BGP. Such attacks have the potential to affect many Internet users and are ideal for denial of service attacks.

Possible solution(s). In the current Internet, the possibility of BGP attacks and misconfigurations has been so far mostly dealt with "Best Common Practice" documents from router vendors. Such documents typically recommend practical measures to prevent a router from being hijacked, and to avoid fake or incorrect advertisements from being accepted by a router. Also, routers are often protected using the MD5 signature option to prevent the TCP connection from being spoofed or hijacked. Even though these countermeasures are used in practice, they are unable to deal with compromised routers. Hence, advanced techniques are required to protect routers against possible compromise.

There are some academic proposals to tackle some of these problems. These proposals focus on the protocol side of router protection, for example, so-BGP and S-BGP could be employed to address routing attacks, but both are considered too expensive by operators.

3.3 IPv6 and direct reachability of hosts

Threats. The Internet Protocol version 6 (IPv6) is the next generation network layer protocol for the Internet. The main motivation for the design and implementation of a new version of such a core Internet standard is the upcoming exhaustion of the IPv4 address space. An ongoing survey [55] currently projects that IANA's unallocated address pool will be exhausted in March 2011, and that regional registries' unallocated pool will run out a year later. Unfortunately, there is little economic incentive to deploy IPv6 before address exhaustion. Furthermore, because of "network effects," an IPv6 deployment is of little use until the rest of the Internet has also deployed IPv6. It is therefore hardly surprising that adoption of IPv6 has been slow. According to a recent Google study [46], less than 10% of the users had functional IPv6 connectivity as of October 2008. Nonetheless, rapid and widespread deployment of IPv6 will become inevitable once the IPv4 address space is exhausted.

Transition Issues. A sudden transition to IPv6, triggered by the unavailability of IPv4 addresses, may well exacerbate the security risks that are unavoidable in such a major upgrade of networking infrastructure. Furthermore, the transition phase itself carries its own risks. In the span of time in which IP versions 4 and 6 will coexist, network administrators will face the complex task of policing both protocols, as well as their interactions, such as the use of tunnels to send IPv6 packets over IPv4 (6to4) or UDP (Teredo). As an example, tunneling of an IPv6 packet over IPv4 could be used to avoid firewall restrictions or inspection from an intrusion detection system. Security analysis of these transition mechanisms has shown novel threats enabled by both 6to4 [132] and Teredo [61]. These include new avenues for denial of service attacks, and a greater ease in performing address spoofing. The inherent complexity of the transition phase, coupled with the lack of knowledge on IPv6-related security issues on the part of network administrators, may well mean that IPv6-related misconfigurations will be one of the primary avenues of attack for Internet criminals.

Universal Addressability. Like IPv4, IPv6 has been designed to provide universal addressability for all devices on the Internet. However, the scarcity of IPv4 addresses has led to work-arounds such as Network Address Translation (NAT) that allows for machines with only a local IP address (that is not globally unique) to communicate with the rest of the Internet. While the use of local addresses and NAT was not originally a security measure, it effectively provides a very restrictive firewall that allows no incoming connections to the devices behind NAT. The IPv6 address space, on the other hand, is easily large enough to allow all Internet-connected devices to have globally-unique addresses. This has numerous technical advantages and simplifies the development of new network applications. On the other hand, if the ingress filtering provided by NAT is not replaced by an appropriate firewall, a large number of home and corporate hosts that were previously exposed only to client-side vulnerabilities (when browsing the world wide web or using other applications) will suddenly also become a potential target for server-side attacks.

Topological Scanning. Another consequence of the huge IPv6 address space is that it will not be possible to perform a brute force scan of the IPv6 address space by simply sending packets to all (or many) addresses on the Internet. Brute force scanning will not reveal all hosts on a network to attackers or allow Internet worm to spread rapidly. Nonetheless, other techniques may well successfully achieve these same goals. The presumed secrecy of IPv6 addresses should not lure network administrators into a false sense of security. Recent research has used mathematical models to explore the propagation of self-replicating network worms in a hybrid, IPv4 and IPv6 network [162], as well as in an IPv6 only network [62]. The fundamental problem is that the IP address of a host is not really secret because it has to be known to any other host it communicates with. As an example, an attacker that has control of a single network node can quickly learn the addresses of all other nodes it communicates with, for instance, by reading the DNS cache. Furthermore, if he is able to sniff (even encrypted) traffic between other nodes, he can harvest their addresses. This type of topological scanning may well allow more sophisticated Internet worms to spread quickly even in the future IPv6 Internet. Furthermore, any Internet service to which a user connects can harvest the user's address. Thus, we can imagine that Internet criminals would buy and sell IPv6 addresses just like they currently buy and sell email addresses to use as targets of phishing and spamming campaigns, especially in countries with lax privacy regulations where such commerce may well be legal.

Additional Issues. Specific features included in IPv6 may also cause security problems. IPv6 Routing Headers have been shown to be an extremely useful tool for attackers, allowing them to amplify their denial of service attacks and to perform advanced network discovery [123]. For this reason, IPv6 Routing Headers are in the process of being deprecated by the IETF [4]. The network auto-configuration features of IPv6 may also pose a security risk. As an example, an attacker with a foothold in a network may attempt to use ICMPv6 Router Advertisement messages to establish a rogue router, re-route legitimate traffic through it and perform a man in the middle attack.

Possible solution(s). The EU has started initiatives and is trying to push IPv6. Currently, there is no large need for IPv6. However, in the near future, as the IP address space will not be sufficient to connect a large number of devices, IPv6 will become inevitable. Just like there were problems with the implementations of IPv4 in the early days of the Internet, we will probably face IPv6-related implementation issues and vulnerabilities. This time, however, the attackers are more organized and aim to make illegal financial gains. As a result, awareness needs to be raised among ISPs as well as industry about the potential threats that will arrive with IPv6. Vulnerability analysis tools that have been used to improve IPv4 stack implementations need to be adapted for IPv6.

An early, gradual adoption of IPv6, combined with IPv6 training of network administrators and engineers, can avoid the high security exposure that we expect would be associated with a last-minute scramble for IPv6, deployed as a reaction to IPv4 address exhaustion. The security of the future IPv6 network will also depend on which of a plethora of standards and proposals for IPv6 extensions [57] will see widespread adoption. The operational and research security communities need to be involved in this transition, to make sure that the real-world deployment of IPv6 improves, rather than worsens, the security of the Internet.

3.4 Naming (DNS) and registrars

Threats. For a long time, the domain name service (DNS) was considered to be a reliable workhorse with a single, simple task: map human-readable names to IP addresses. However, this has significantly changed in the last couple of years, and as an emerging threat, DNS is now routinely abused in malicious operations on the Internet. In particular, fast-flux DNS is a recent technique which overloads the A (address) records in the DNS server. One A record will have multiple IP addresses, making it redundant: each client will try one IP address after another, until it can successfully establish a connection.

In botnets, fast-flux techniques are used to connect to and hide the command and control servers (C&C). That is, it is used by botnets to hide phishing and malware sites behind a changing network of compromised hosts. These sites are used to deploy malware (botnet software such as Storm Worm) to unaware users. It can also refer to the combination of peer-to-peer networking, distributed command and control, web-based load-balancing and proxy redirection used to make malware networks more resistant to discovery and counter-measures.

The fast-flux networks are a group of compromised (hacked) computer systems that have a public DNS record. These records change very fast, which makes the detection of these networks harder. Fast-flux networks have multiple (tens, hundreds, or thousands) of IP addresses assigned to them. These IP addresses in the A records are changing very fast, using round-robin IP addresses and a very short TTL. An unaware user connecting to a website might be connecting to a different infected host each time. The IP address pool is usually not the final destination. Instead, these hosts merely serve as redirectors that forward the requests to other backend servers (that provide the content). This technique is typically used for load balancing and high availability, but botnet herders have adapted this approach for illegitimate purposes. The controlling elements are called "motherships" [52]. These motherships are hidden by the front-end fast-flux nodes. The motherships host both the DNS and HTTP services, and can be configured to manage thousands of domains simultaneously on a single host. The motherships provide the information for DNS to the front-end, which then forwards it to the infected client. There are two different types of fast-flux networks: single-flux and double-flux.

Fast-flux networks are a major menace as they are difficult to take down. The problem of fast-flux networks is expected to grow in the future, resulting in a increasing interest by various security groups [53]. In [51], the authors study the significance and general principles of this kind of network service. Their experiments show a rapid increase in the use of this new technology — while earlier measurements from December 2006 [8] showed only very few online fraud campaigns to be hosting content using multiple IP addresses, their evaluation of spam-trapped data from August 2007 suggests the use of fast-flux services to be almost as high as 30%. This trend continued in last two years.

From the collected data, analysts were able to extract three key features common to fast-flux service networks. First, the number of unique `A records` returned by a DNS lookup is significantly higher than observed for legitimate hostname lookups. Alike, FFSN employ more nameserver entries than casual networks. Last, as the single stepping stones are typically widely distributed over many Internet service providers (ISPs), the number of ASNs is rather high (whereas legitimate networks are usually hosted within a single system).

Based on their findings, Holz et al. propose a tool to detect networks that employ fast-flux services by calculating a `flux-score` based on observed values of the three characteristics described above. The authors of [120] extend this idea and present `FluXOR`, a tool to detect and monitor botnet networks that employ fast-flux services. FluXOR achieves its goal by monitoring a (potentially) malicious network from the perspective of a victim that is lured into accessing a resource provided by a webserver hidden behind a proxy, i.e., gathers data from `outside` the infected network.

More precisely, the tool continuously monitors (queries for) the IP addresses associated with suspicious hostnames. It then tries to extract and identify multiple features that can be used to distinguish the observed networks from casual, benign services employing load balancing or the use of mirrors to provide content.

In addition to the features presented by Holz et al., these new features mainly concentrate on TTL values associated with DNS records, the domain age and used registrar, as well as more precise ways of measuring the heterogeneity of botnet clients. The values obtained by FluXOR for a given hostname can then be compared to a training set of manually classified networks. This allows to decide if a network is hosted by a fast-flux service without having detailed information on the network's internals. A similar system that provides real-time monitoring and statistics is presented in [21].

In [108], Nazario and Holz extend work from [51] and provide more detailed information about lifetimes, sizes, and separability of fast-flux service networks. By inspecting data collected over 4 months in early 2008 they gathered that FFSN have an average lifetime of 18.5 days. The largest and longest-living network operated almost 60 days, spanning over 100,000 network nodes. By clustering for distinct set of IP addresses, the authors identified 26 distinct fast-flux botnets providing different services such as ”pharmacy product” stores and sites used in phishing attacks.

In contrast to other work, Konte et al. [72] study fast-flux techniques by assigning a large set of networks to 21 distinct scam campaigns exhibiting fast-flux behavior. They investigate on common properties among and unique characteristics within the individual campaigns and compare the results to a large set of benign networks. The authors further measured the network's dynamics, i.e., the rates at which DNS mapping are changed, the speed at which the number of network nodes grows, as well as the rate of changes in the DNS hierarchy (for double flux systems). Their findings suggest that the network's dynamics pose another means of identifying malicious networks in terms of fast-flux techniques.

Possible solution(s). Domain registrars seem to be very lax when it comes to registering and selling domains. To combat fast-flux networks, cooperation by ISPs and domain registrars is inevitable. Also, ISPs and domain registrars need to be held responsible, to a certain extent, for the damage that is caused by the services that they host. For example, if a domain registrar does not disable a registration after repeated complaints, there needs to be a legal mechanism that can hold them accountable. Currently, domain registrars are often lax and are not concerned as they are not liable.

3.5 Wireless communication

Threats. Heterogeneous wireless networks hold the promise of empowering people through a digital environment that is aware of their presence and context, and sensitive to their needs. These wireless networks will enable application areas such as ubiquitous/pervasive computing, resiliency and quick recovery from nature and man-made disasters, and provision of safety services for a better quality of life for elderly and disabled people. Specific applications that make use of the capability of wireless communication systems to connect the physical world to the cyberworld range from monitoring bridges, roads, tunnel structures, and water quality, to controlling the temperature of our homes according to the presence and location of people.

Wireless communications also offer many convenient advantages compared to traditional wired communications within the industrial domain. Today, wireless communications are not yet widely used in practice in industrial environments. Most plants are only considering them for information gathering in the form of measurements, but not for closed-loop control [65]. However, wireless technology is compelling because of its many advantages: operator mobility, safety by enabling remote access to noxious environments, access security for visualization and optimization, and the immediate benefits of their deployment [18]. There are already applications for maintenance, condition monitoring, asset management, asset tracking, etc. Such applications improve efficiency but may not be directly related to actual control or incremental measurement of processes. Based on these compelling advantages greater adoption of wireless communication in industrial control can be expected, thus with an overall growth in its market share.

Experts from WINA and ISA [87] predict that within 10 years, even critical control communications will be wireless. Recently, following the WirelessHART and ZigBee Alliance announcements and after approving the SP100 standard for industrial wireless communications by ISA, there is already use of wireless communications in industrial and even critical applications. Despite this, the single industrial wireless standard ISA-SP100.11 does not give enough guarantees for dependability and security to critical systems and applications. It can be expected that the use of such systems and hence the problems will expand in the future.

One main security aspect of the wireless communications in general follows from the unbounded nature of radio frequency (RF) propagation. The perimeter of a wireless network cannot be limited and controlled as can be done with a wired network. There are reflected signals, which find their way out of buildings. These dispersed signals could be detected by motivated attackers that could then attempt to interfere with them if they are in physical proximity of the facility. Thus, traffic can be passively captured and an attempt to penetrate the network could be made with the aim to reach other connected enterprise networks.

The strict resource constraints of wireless networks (i.e., radio frequency bandwidth, energy), and other characteristics of these systems, such as mobility and shared broadcast medium, require the use of complex control mechanisms to conserve system resources. This makes these control mechanisms a target of choice for denial of service attacks. We have recently seen that most wireless networks are sensitive to what we call cross-layer attacks. Such attacks focus on specific frequency carriers, at specific instants of time, with the objective to corrupt critical control messages crossing multiple layers. With very little resources, a smart attacker can cripple a complete wireless network.

Such attacks can consume four orders of magnitude less energy than previously known attacks. It is shown that these attacks apply to various forms of cellular networks (e.g., GSM, 1xEvDO, WiMAX), wireless local area networks (e.g., IEEE802.11), but also MANETs.

Another serious security problem in using wireless communication is related to the security of the access and communication protocol itself. Here, we will face the same type of problems for wireless applications as we already see for non-wireless ones.

Possible solution(s). Whereas all of the experts are convinced of the extended use of wireless communication for both private and industrial communications in the future, some of them also comment on the risks involved, and especially emphasize the careful introduction of these technologies. The first and main consideration when addressing security of industrial wireless communications is the conformity to the ISA-SP100 Usage Classes. There are many useful recommendations like those in [90, 91], where detailed recommendations for securing wireless networks are given.

Some of these considerations for industrial environments can be as follows: depending on the problem, use of least susceptible frequency band in case of intensive electromagnetic interference (EMI), or increasing the transmit power level by using a higher-gain antenna, if the amount of electromagnetic noise is significant. In other cases, it may be better to reduce transmission power levels or deploy directional antennas in order to reach negligible levels of the stray signals.

The use of a *frequency hopping* (FH) radio with configurable hopping channels and patterns can help mitigate/avoid interference, reduce multi-path fading, as well as provide an additional measure of security, if a non-default hopping pattern is used and also changed on a periodic basis [90, 91].

The IEEE 802.15.4 standard [56] supports an optional Guaranteed Transmission Service (GTS). To mitigate real-time operation problems, it is recommended to use *guaranteed transmission mode*, whenever possible. For securing the industrial wireless communication the secure mode is supported in the standard.

It is shown that cryptographic randomization, agility, and diversification, in a game-theoretic context can provide the tools for building resilient wireless networks against both external and internal attacks. Such techniques can even allow the identification of internal attackers.

3.6 Denial of service

Threats. In this section, we outline attacks that are possible against the network infrastructure (e.g. routers) and that aim at causing wide-spread denial of service. The core Internet infrastructure is a high-value target from the malicious user perspective and will be subject to targeted attacks. These attacks are not new, but we expect to see their number rising in the years to come. Furthmore we expect existing attacks to target new applications that are being deployed on the Internet like VoIP, Internet TV, etc.

Direct attacks against routers are already commonplace, albeit not openly discussed. The tendency is towards worm-based exploitation of home routers, wireless access points, and similar - typically badly secured - networking equipment. These types of attack allow for knocking out network access, but also enable more sophisticated man in the middle attacks. Emerging threats include DNS or DHCP highjacking, with potentially serious security implications. (Example: "Symantec warns of router compromise," `www.news.com`, 24. Jan 2008)

The physical layer and, in general, lower layer attacks are also of increased significance. Where physical access to fibers or networking equipment is available, many attack forms are possible, including wiretapping and router intrusions. Social engineering attacks are often successful in bypassing physical access control mechanisms. These attacks require more effort than remote attacks, but where the value of information on the Internet is increasing, this type of attack will become more popular.

Attacks will also expand into higher-level networking services. As the TCPIP layers are becoming increasingly robust and attack-resistant, attacks will not only move to the lower layers, but also to higher layers such as the application. DNS poisoning attacks (various forms) also fall into this category as discussed in a previous subsection. Internet infrastructure is directly or indirectly also affected: Networking equipment is becoming increasingly more complex, and application layer attacks will also be seen against the network itself.

Possible solution(s). On the purely systems side, where the operating system of routers is attacked, one could possibly adopt solutions from generic operating systems. Equiping our router, and network equipment operating system in general, with security software (e.g. antivirus, etc.) could go a long way toward protecting such critical equipment.

On the resource side, where the attacker employees flooding attacks against the routing and networking infrastructure, things become very difficult. In such cases, one could adopt replication of hardware and replication of routes for fault tolerance to survive this type of DoS attacks. There is currently no deployable, easy solution for routing security. This could lead to major Internet outages, and even a "split" of the Internet.

Chapter 4

Threat Category: Hardware and Virtualization

4.1 Overview

Hardware-based attacks and attacks against virtualization infrastructure are two areas that we believe will be increasingly gaining importance in the future with respect to emerging threats. Because of cost considerations, hardware production is increasingly relocated to cheaper countries. As a result, there is an increasing risk that the devices that are produced in these countries cannot be fully trusted.

At the same time, hardware virtualization technologies such as VMware also increase in importance and popularity. Again, cost arguments push businesses to make more of their available resources and share the same physical hosts between many virtual machines. In the last consequence, virtualization leads to a situation where hardware is no longer hosted in-house but purchased from third-party providers. That is, virtualization is especially important as a crucial component in the emerging cloud computing arena. Hence, we expect novel threats to emerge in this area.

In this section, the hardware and virtualization category, we cover threats due to new hardware and software developments that allow computation to be moved to virtual computers, and ultimately, the cloud. It also covers malicious hardware.

4.2 Malicious hardware

Threats. Increasingly, hardware design and fabrication has come to resemble that of software: hardware logic modules (resembling software libraries) are licensed from third parties and combined in designs of greater complexity, while the fabrication is outsourced to a low-cost manufacturer or otherwise off-shored.

While this new way of constructing hardware has brought great benefits in terms of design reuse, rapid development and prototyping, and lower component and product costs, it has also introduced new vulnerabilities for high-value or sensitive users of such technologies. In particular, a sufficiently motivated adversary (or a disgruntled employee) can introduce backdoors (*Hardware Easter Eggs, or HEEs*) during the hardware design or fabrication phases. For instance, a hardware designer, by changing less than ten lines of Verilog code, can easily modify an on-chip memory controller to send data items it receives to a shadow address in addition to the original address. Such HEEs can be used in attacking confidentiality (e.g., by exfiltrating sensitive information), integrity (e.g., by disabling security checks such as memory protection), and availability (e.g., by shutting down the component based on a timer or an external signal). HEEs cannot be detected using standard state-of-the-art pre-fabrication testing techniques because the attacker is likely to delay enabling or opening the backdoors until after deployment using simple control circuits. It is even possible to create low-gate-count general-purpose HEEs that can be leveraged by attackers to launch a variety of powerful attacks against the system.

The threat is clearly realistic. In fact, we encountered some first manifestations during the project life time. For instance, The US National Counterintelligence Executive, Joel Brenner, in 2008 announced that an organized crime group succeeded into tampering with commercial credit card readers which were shipped to and installed in retail stores around Europe. These devices are believed to have been in use for at least nine months before they were discovered and the fraud amounts to millions of euros.

Because hardware components (including embedded HEEs) are architecturally positioned at the lowest layer of a computational device, it is very difficult to detect attacks launched or assisted by those components: it is theoretically impossible to do so at a higher layer, e.g., at the operating system or application, and there is little functionality available in current processors and motherboards to detect such misbehavior. The state of practice is to ensure that hardware comes from a trusted source and is maintained by trusted personnel: a virtual impossibility, given current design and manufacturing realities. In rare circumstances, when volumes are relatively low and the risk is high, physical inspection and verification of the hardware may be conducted. Such inspection is destructive, costly, and time-consuming, and thus can only be applied in few cases and for a small number of samples.

Establishing trust in the hardware components underlying all modern IT will likely prove a key future challenge for the security and hardware design communities. While HEE-based attacks are virtually unheard of to date, economic, technological, and social drivers make these attacks more likely than ever before, while the potential damage from such an attack is extremely high: shutting down an hypothetical adversary's cyber-infrastructure (or "just" a significant or sensitive part of it) in the event of an armed conflict or during a period of diplomatic tensions can be an effective and cheap way of forcing the outcome.

http://news.softpedia.com/news/Hundreds-of-Tampered-Chip-and-Pin-Devices-
shtml

Possible solution(s). Addressing the problem requires a concerted, long-term effort in physical design and manufacturing methodologies, secure and trusted fabrication practices and operations, post-fabrication testing and verification techniques, and runtime HEE detection and mitigation. The problem domain represents both challenges (in terms of the physical parameters, low-level of abstraction, ease of implementing certain catastrophic attacks, and lack of access to IC internal state) and opportunities (the ICs interface to the rest of the environment is limited and can be completely controlled). We believe that a combination of techniques, combined with updated manufacturing practices, can help mitigate the risks at acceptable cost, both in terms of research expenditures and manufacturing/operational practices.

4.3 Virtualization and cloud computing

Threats. In the last ten years, the popularity of virtualization has increased significantly. Virtualization is the method by which a ”guest” operating system is run under another ”host” operating system, with little or no modification of the guest OS. In 2005 and 2006, extensions to the x86 architectures by Intel and AMD made virtualization easier. In particular, it is possible to “outsource” computation to remote machines (such as the “cloud”). As a result, the perimeter between local data and computation is increasingly blurred, and potentially sensitive data is moved around on the Internet.

Virtualization is popular because it makes the maintenance of computing systems easier. Furthermore, virtualization techniques are increasingly being used in the analysis of security threats such as malware. We believe that virtualization technologies will be increasingly attacked in the future. For example, the attackers will be interested in finding techniques to break out of the the virtual guest in order to infect the underlying host. In a recent paper [157], a proof of concept of such an attack was demonstrated. A flaw in the XEN virtualization environment allowed the authors to gain control of a host machine even avoiding additional security measures installed on the host system. Such an attack could be easily launched on a large scale by breaking out of a virtual machine hosted on online cloud services such as Amazon EC2. Here, a single vulnerability in a virtual machine could allow an attacker to simply rent more virtual machines, using his exploit to break out of each new, virtual host, and as a result, successively taking over the complete cloud.

Some security researchers have discussed the possibility of a “Blue Pill” attack, using a virtual rootkit similar to the one created by security researcher Joanna Rutkowska. This kind of rootkit, in theory, can hide in the hypervisor and away from the reach of today’s security tools. Although blue-pill-like attacks have not emerged so far, we believe that stealthy malware that uses virtualization is a real threat that will emerge whenever the attackers see the need for it. That is, if security tools improve and can deal with techniques such as obfuscation (e.g., using behavior-based detection), then there will be a need for more stealthy malware.

Perhaps more problematic is the fact that intrusion detection tools cannot be deployed today to look at inter-VM traffic. As a result, as virtualization technology is used more and more to host services, the intrusion detection tools of today will be rendered increasingly ineffective.

Separation between virtual machine instances is another security problem. Even though, theoretically, one VM instance should not be able to inspect other virtual images running on the same host, it might be possible for an attacker to infer useful information about other instances using a side channel attack. As the hardware is shared between the instances, timing attacks like the one presented in [71] might be used to recover information about cryptographic secrets used in another virtual machine.

Also, note that malware samples are increasingly becoming virtualization resistant. That is, many malware samples have built-in checks and are testing for the presence of virtualization. If they realize that they are running in a virtual environment, often, they change their behavior. The aim of these malicious programs is to prevent the malware analysts from understanding their inner workings by running them in a virtual environment. Virtual environments are often used to analyze malware and clearly, malware authors are aware of this fact.

Possible solution(s). Awareness needs to be raised in industry to change the common belief that virtualization techniques are perfect for security. Furthermore, research is needed to make virtual malware analysis environments more resistant to evasion techniques. Virtualization providers need to provide hooks into the inter VM communication channels so that intrusion detection systems can monitor the traffic and detect attacks and also current malware detection applications have to be adapted to the new threat environments by applying for example approaches like the one presented in [121] which allows active monitoring of running VM instances.

Chapter 5

Threat Category: Weak Devices

5.1 Overview

Weak devices are devices that are restricted, either because they have computational limitations or because of limited battery power. Examples are embedded computer equipment, sensors, displays, phones, and PDAs. Ordinary environments are being enhanced by interconnected, weak devices. Compared to the world of personal computers and servers, security in weak devices and sensor environments is more difficult to analyze because fewer security problems have occurred so far.

On the surface, studying security in such environments seems to require a crystal ball and/or a lively imagination. While we do not deny the usefulness of either, we argue that studying existing threats and trends allows researchers to form a coherent picture of (at least some of the) threats likely to emerge in the future.

For instance, we observe that mobile phones are becoming like computers with full-blown operating systems, lots of applications and (as a result) many bugs and vulnerabilities. In other words, we see increasing opportunities for hackers to attack phones. Still, we do not see many attacks on phones (yet). It is more lucrative for an attacker to hack a PC than it is to hack a phone. After all, PCs are used to enter credit card details, passwords and many other interesting potential targets that all represent value. Phones are mostly used to, well, phone people.

As a result, attacks on mobile phones are still relatively rare. However, we witness a trend that phones will be used increasingly for financial transactions, in addition to normal, PC-like, Internet access. This makes them a more interesting target for attackers. Combining the two trends – increasing opportunities and increasing incentives – allows us to predict that smartphones are much more likely targets in the future than today. Under the same token, other weak devices that are being deployed and interconnected will become the targets of tomorrow.

In the following, we first discuss emerging attacks against sensors, which are the prototype of weak devices. Then, we specifically look at the threat of malware on mobile (weak) devices.

5.2 Sensors and RFID

Threats. The convergence of control with communication and computation will make sensor networks the new dominant "computing class." This class will provide the ability for large numbers of sensors, actuators, and computational units (interconnected), to interact with the physical environment. This computational shift is going to bring a big shift also on computer security issues.

One problem is that small sensors require a means to communicate. This is typically a wireless connection. However, in addition to the security concerns of wireless networks in general (previously discussed in Section 3.5), wireless sensor networks have a number of additional issues:

- The nodes in sensor networks are in general very limited in terms of battery, storage, and computational power. Therefore strong cryptography and other general security tools are of limited use, if at all available. An attacker can have much more powerful hardware than the nodes being attacked.
- Sensor networks typically reside in unattended environments where an attacker can physically destroy nodes, add malicious nodes or in other ways tamper with the hardware of the network.
- Nodes in a sensor network die for many different reasons. For example, batteries can run out, nodes can break during deployment when they are deployed (thrown out from an air plane) or break during operation due to a harsh environment. It is hard to distinguish such natural failures from a malicious attack where nodes are deliberately destroyed.

There are many venues of attacking sensor networks [22, 122], including the following.

- Snooping information
- Inserting false or misleading information [89]
- Jamming radio channels
- Making nodes run out of battery by never letting them sleep
- Giving the impression of phantom nodes that do not exist [110]
- Giving the impression of connectivity that does not exist [67]
- Making messages go through an attacking node that can selectively drop messages from the system [67].

A special class of sensors is based on Radio Frequency Identification (RFID) technology. What makes this class particularly interesting is that the technology is pushed quite aggressively by industry to help create what is known as an Internet of things. RFID tags contain a tiny, miniaturized chip that is powered by means of induction. Their low cost make it possible to attach them to almost everything: key cards, public transport tickets, clothing, products in a supermarket, pets, passports, and just about anything else. Mostly they are used in supply chain management to identify products. They may also be used to identify users. In these cases, the tags typically contain a fixed code. However, they may be used to store and process a small amount of information also.

RFID introduces a host of security threats. It has been argued in [133] and [149] that RFID technology threats span much of what is known as the STRIDE threat model (originally proposed by Microsoft), which includes Spoofing identity, Tampering with data, Repudiation threats, Information disclosure, Denial of service, and Elevation of privilege.

Much of the discussion about RFID security to date has focused on information disclosure and tag replication. For instance, various versions of the Mifare chip, which is used extensively in public transport, have come under attack when it was shown that the protective methods (including the encryption) can be broken easily [42]. The findings have had tremendous and quite costly consequences in many countries. For instance, they have jeopardized the introduction of the public transport card in the Netherlands altogether.

Besides the above problems with a particular chip, we suggest that other threats should be taken more seriously also. Spoofing and tampering are particularly worrying. Spoofing can be accomplished in two ways: attackers may spoof the identity of an RFID reader (in which case unauthorized scanning may be performed), or attackers may spoof the identity of the RFID tag (which may, for instance, lead to unauthorized access). Since tags are typically designed to be as cheap as possible, it is questionable whether high-grade authentication will always be applied in practice. Tampering occurs when attackers modify a tag. For instance, an attacker may modify a tag that links the user to criminal or terrorist activities. The inverse is also possible, where a criminal modifies a tag so as to appear as a citizen in good standing. Modification of tags in the supply chain may disrupt business operations, or in the case of price tag modification, may lead to loss of revenue. However, tampering may also mean the addition or removal of tags. Adding tags to a shipment may make the shipment appear to contain more items. Deleting tags may render items (e.g., products in a supermarket) undetectable.

A development that we consider interesting but not at all surprising, is that RFID tags can be used to carry and distribute malware [130].

A selection of the above threats have, in one way or another, already been discussed in existing literature. In our opinion, they *all* need to be looked at carefully, which in turn requires careful tag management (who is allowed to read or write which tags and when?). The threat or challenge is also related to scalability (see a later discussion on scalability in Section 6). Users will not even be aware of all the tags they own and carry around. How can we make sure that the appropriate access policies are applied to things of which we are not aware?

Possible solution(s). Tag management is crucial in all security aspects related to RFID. Some researchers have proposed a guard or blocker device carried by the user that allows users to block readers from communicating with the tags in their possession [64]. All access may then be mediated (vetted) by a security device that makes sure that unauthorized access is not permitted. The problem with such solutions is that it may be difficult to apply specific policies to specific tags. Even the simpler scenario of applying a policy to all of the tags of a user may be problematic, as the user may not be aware of all the tags he/she owns. Worse, it is even harder to apply some policy when tags belonging to other users are in the vicinity. Typically, RFID blockers can prevent readers from interacting with all tags in the user's vicinity. However, the user may not own all tags in the vicinity. We may need to distinguish between important "known to be owned" tags (say the tag in user's passport) and less important ones that may or may not be owned by the user and differentiate access control accordingly.

In summary, we consider the following three approaches worth further pursuit in the context of sensor devices.

- Autonomic solutions where the system will continuously evolve and control its security.
- Solutions that will mask subsystem takeover.
- Combining sensor information with physical information for verifying certain operations.

5.3 Mobile device malware

Threats. Carrying a so-called smart mobile phone is almost like having a powerful computer today. In fact, smart phones that are sold today are as powerful as desktop PCs that were sold ten years ago. An increasing number of phones sold today include extensive online access, keyboards, and other typical computer functions. However, the power and convenience comes at a cost. Just as traditional computers face security threats, so do these mobile devices. Unfortunately, the larger the functionality becomes that these devices support, the more vulnerable they become to attacks. In the near future, it is highly probable that these devices will become susceptible to the same type of threats that plague our laptops and desktops, and in particular, malware.

The most common operating systems used by mobile phones and personal digital assistants (PDAs) are Microsoft Windows Mobile and Symbian. Windows Mobile 2003 and Windows Mobile 6 are based on the Windows Mobile platform, which has a shared-source kernel strategy. Because of Microsoft's developer-friendly environment and shared-source policy for Windows Mobile, more phone manufacturers have begun to adopt it. At the same time, these same features are expected to attract more and more malware writers. We believe that malware for mobile devices should be an increasing concern for researchers and industry.

First, unlike normal PCs, smartphones run on battery power. Power in mobile phones is an extremely scarce resource. For instance, one of the main points of criticism against Apple's iPhone 3G concerned its short battery life [100]. Software developers bend over backward to make core code run fast on phones, because every cycle consumes power, and every Joule is precious.

As a consequence, many of the security solutions that work for desktop PCs do not suit smartphones as they are too heavyweight. File scanning, taint analysis, and system call monitoring all consume battery power. Battery life sells phones, and consumer hate recharging. The likely result is that both vendors and consumers will trade security for battery life.

Second, unlike traditional computers, phones go everywhere we go. Attacks may come from sources that are extremely local (e.g., via Bluetooth). A person with a laptop or another smartphone that happens to be in the same room could be the source of an attack. That means that security solutions based on in-network scanning are useless: they will never even see the bytes that are used to take over the phone, steal information, and plunder your bank account.

Worse, phones are small devices, and we do not always keep an eye on them. We may leave them on the beach when we go for a swim, slip them in a coat or shopping bag, forget them on our desks, etc. Theft of a phone is much easier than theft of a desktop PC or even a laptop. Moreover, attackers could "borrow" the phone, copy data from it, install back-doors, etc. This is an important difference with the typical desktop PC.

What would it buy an attacker to steal a mobile phone (and perhaps return it later)? Well, having the device in your hands opens up a wide range of options for attacking the device that would otherwise not exist. Hardware attacks, for instance [9]. Attackers may use hardware debugging equipment to snoop on data traveling from and to memory, read or write keys, etc. Direct loss of private data may be an immediate result. However, another and perhaps more insidious threat is when the phone is returned to the owner with a backdoor that allow attackers to gather information for a long period of time.

Is this practical? Let us have another look at the example of bluebugging; the faulty implementation that made the earliest Bluetooth phone vulnerable to remote exploits was fixed fairly quickly. However, phones could still be compromised. The only thing that was needed was that the attacker talked "the victim into handing over the phone, which the bluebugger manipulates to set up a backdoor attack and then hands back" [82].

Currently, although there are known attacks against mobile devices [31, 32, 142], miscreants have not been targeting these devices on a large scale. This is probably because attacking traditional computers is easier and more profitable currently. However, as users will increasingly use their mobile phones to surf, make purchases and communicate sensitive data, these devices will become interesting targets for attack. For example, some mobile network providers let their customers transfer money from their mobile accounts to other customers via SMS messages.

Recently, a malware targeting mobile devices was discovered [68] that automatically transfers money to the attacker via such functionality.

We believe that mobile malware can be used to steal sensitive financial information if mobile devices become widespread financial instruments. Also, we envision that these devices will be used in the future to track users, probably listen to their conversations (e.g., by remotely turning on the microphone), and blackmail individuals. It is also possible that mobile devices will become part of botnets once mobile Internet access becomes cheap and ubiquitous. In fact, mobile Internet access prices have been steadily decreasing in many EU countries.

Mulliner et al. [104] identified the integration of different communication techniques as a potential threat to a user. For example, integrating a GSM modem with a Wi-Fi network component into a single mobile device might open the user to the threat of malware that can cross the border of the individual transmission techniques. Such malware could infect the device through a vulnerability of the Wi-Fi component and subsequently dial or send text messages to premium-rate numbers via the GSM modem.

In [102], Mulliner describes and implements attacks against current near field communication (NFC) enabled mobile phones. NFC today, is mainly used for mobile payment and ticketing. By modifying the information broadcasted by a NFC tag the authors succeeded in performing different attacks against NFC enabled phones. These attacks include denial of service and man-in-the-middle attacks. An attacker can perform URI spoofing with tags used for ticketing services to lure a user into calling premium-numbers instead of the legitimate ticketing number. Furthermore, the paper discusses and introduces a proof-of-concept NFC worm.

How malware targeting mobile devices propagates in mobile phone networks modeled and simulated in [37]. The propagation of malware that relies on other means of mobile communication and infection (e.g., messaging, Bluetooth) is in the focus of [20].

We stress that the trends are not working in our favor. On the one hand, mobile phones are an increasingly attractive target for attackers. On the other hand, because of power limitations and physical exposure to hostile environments, phones are inherently more difficult to protect than traditional computers. In a future Internet, it is imperative that solutions are found to protect mobile devices that carry valuable data. Existing paradigms, based on in-network scanning and/or traditional anti-virus software cannot be simply ported to mobile phones.

Possible solution(s). Current anti-malware solutions need to be adapted so that they can be used to detect and respond to mobile malware. Some work has already been going on in this area. However, the proposed solutions are still heavy weight and mobile devices still have performance and bandwidth problems.

Network service providers and GSM companies need to start thinking about how to mitigate the threat on the server-side, before it reaches the end users. To this end, Becher and Freiling propose a mitigation strategy [17] where any application that is about to be installed on a mobile device is first transmitted to the network provider for analysis. This analysis executes the sample in a restricted environment (i.e., a sandbox) and monitors and records the behavior of the sample in terms of API calls. Such an analysis scheme closely resembles techniques applied by TTAnalyze [16] and similar tools on PC hardware and operating systems. Only if the analysis does not identify any malicious behavior in the sample it is allowed to be installed.

To mitigate the threat of mobile malware that crosses service boundaries, Mulliner et al. [104] demonstrate an approach that relies on resource labeling. During execution all processes and resources (e.g., files) on a smart phone are labeled with the network interfaces they have been in contact with. Whenever a process invokes a system call, its labels are compared against a global policy file that specifies whether the action should be allowed or not.

The Multimedia Messaging Service (MMS) is used to exchange messages between user agents running on mobile devices. Mulliner and Vigna proposed a system [103] that performs fuzzy testing of such user agent applications to reveal possible vulnerabilities. This approach identified multiple security vulnerabilities that allowed to compromise system security. In addition, they implemented a proof-of-concept attack that exploits one of the detected buffer overflow vulnerabilities to execute arbitrary code received through an MMS message.

It would be interesting to explore whether security can be (almost) entirely decoupled from the phone itself. Any functionality that is applied elsewhere will not drain the phone's battery and may therefore consume more power. A simple solution in that direction is to apply anomaly-based intrusion detection systems in the network [23], but we have already argued that this approach is limited because some of the attacks stem from a local source (e.g., via Bluetooth) and thus never reach the network. A radical alternative model might try to replicate the state of the phone in a dedicated security server [38]. By keeping the copy of the phone in the security server in sync with the phone, all security checks can be applied in the server and not drain the battery of the phone. Yet another approach might be to apply more rigid coding practices on phones that rule out the occurrence of certain exploitable bugs by design. The problem with this solution is that it does not seem plausible that vendors will opt it in the short or medium term.

Chapter 6

Threat Category: Complexity

6.1 Overview

The complexity of networks and systems has increased dramatically over the last years, and the tendency is still growing. This means that operators do not longer understand their network in its entirety. This increasing operational complexity will undoubtedly cause more problems in the years to come, both in accidental operational errors, as well as in deliberate attacks. New mechanisms and algorithms to control and monitor network and system complexity are urgently required.

In this category, we consider both many small systems that are interconnected as well as a single big monolithic system. Given the complexity of the systems, we describe the possibility of *unforeseen cascading effects* and *threats due to scale*. Large systems are also difficult to verify, as discussed in *threats to system maintainability and verifiability* and threats due to *hidden functionality*. Finally, we also consider *threats related to parallelism*.

6.2 Unforeseen cascading effects

Threats. Interconnected systems and networks are difficult to model properly and interdependencies between them can lead to cascading effects that are difficult to foresee. This is due to the inherent complexity of the connected systems. It is claimed that nobody *really* understands a network such as the Internet anymore, nor even many smaller interconnected, heterogeneous networks that have been deployed over the past decades. Further, testing is virtually impossible due to the complexity and scale. In particular, testing is often impossible when the system is connected to a critical infrastructure with real-time requirements.

An important class of cascading effects occurs when, e.g., some section of the Internet is attacked or overloaded to the point of service denial and another (perhaps critical) system depends on that section. Even though the attack was not directed against the critical system per se, it is affected indirectly.

Examples of such knock-on effects occurred in Estonia in 2007 when the country was hit by massive denial of service attacks. Not only did the attacks knock out many systems that were directly targetted, but also many services that depended on those systems, to the point that the country lost 50 percent of its "bread, milk, and gasoline" during the peak of the attack.

Similarly, the great north east black-out of 2003 which was partly caused by a race condition (see Section 6.6) had unfortunate knock-on effects: because the power went down, certain public transport services stopped operating, which meant that employees could not go to work, which in turn meant that the services by these workers could not be provided.

It is clear that dependencies are responsible for unforeseen cascading effects. Unfortunately, dependencies in large networks and systems are very difficult to understand due to their complexity. This applies to a single network infrastructure, but connecting two or more infrastructures will make this complexity grow exponentially.

Even though system complexity is an issue in many areas, some factors related to critical systems make the issue of the complexity extra severe in such environments. First, due to the deregulation of markets, critical infrastructures are often run by different organizations that need to cooperate. These organizations are seldom a single unit, but they are comprised by many smaller units as virtual organizations. A complicating issue is then that part of the system may be governed by proprietary protocols while others use open standards. Different system owners may not trust each other, and different parts of the system are governed by their own safety/security policies.

Give a system X, it may be hard enough to estimate which systems depend on X. However, it may be even harder to determine which systems depend on the systems that depend on X - and so on. Without charting the full chain of dependencies, unforeseen cascading effects will occur, whenever the an important service goes down.

As an illustrative example, in the years before the turn of the millenium governments and industry alike spent enormous amounts of money in order to be able to handle the infamous Y2K problem. Some systems still failed. For instance :

- In Ishikawa, Japan, radiation-monitoring equipment failed at midnight, but officials said there was no risk to the public.
- In Onagawa, Japan, an alarm sounded at a nuclear power plant at two minutes after midnight.

http://www.rferl.org/content/article/1109653.html

Lerner, Eric J. (October/November 2003). "What's wrong with the electric grid?". The Industrial Physicist (American Institute of Physics).

These and other examples can be found on: `http://en.wikipedia.org/wiki/Y2K#On_1_January_2000`

- In Japan, at two minutes past midnight, Osaka Media Port, a telecommunications carrier, found errors in the date management part of the company's network. The problem was fixed by 02:43 and no services were disrupted.

- In Japan, NTT Mobile Communications Network (NTT DoCoMo), Japan's largest cellular operator, reported on 1 January 2000, that some models of mobile telephones were deleting new messages received, rather than the older messages, as the memory filled up.

- In Australia, bus-ticket-validation machines in two states failed to operate.

- In the United States, the U.S. Naval Observatory, which runs the master clock that keeps the country's official time, had a Y2K glitch on its Web site. Due to a programming problem, the site reported that the date was Jan. 1, 19100.

- In France, the national weather forecasting service, Meteo France, said a Y2K bug made the date on a webpage show a map with Saturday's weather forecast as ”01/01/19100”.[21]

However, the point is not that these glitches still occurred. The real point is that we had to spend incredible amounts of money and person years to understand all (or most of) the dependencies. It is unlikely that we will be able to keep track of such interdependencies as time goes on,

Possible solution(s). What we need are new, more appropriate modelling tools and an overall better, probably structured and hierarchical, architecture with a security baseline. Removing the human from the loop and introducing automation may help, because the seemingly intuitive action scripted in automated systems might be completely wrong in certain systems and lead to large problems. For essential services, it is important that dependencies should be tracked from the design phase onwards.

6.3 Threats due to scale

Threats. While the Internet has grown to a 100-million node network, our models and intuition of the world has hardly moved from the familiar two-node client-server model. As a result, networks are increasingly vulnerable to attacks such as puppetnets where minor vulnerabilities in the design of the Web can be amplified in proportion to the number of clients and servers in the system. Similar patterns are exposed in metro-area WiFi networks [7], where minor vulnerabilities are amplified into serious threats due to deployment density, and how the change from wired to wireless brings long forgotten vulnerabilities such as DNS spoofing back to the spotlight.

Moreover, as most end users run a version of Microsoft Windows on their desktop and laptop computers, we may refer to a large part of the Internet as a software mono-culture. Unfortunately, in large mono-cultures, faults, however small, tend to generate large problems. An example of such behaviour was identified in 2003 by CAIDA researcher kc Claffy. In a report, titled “Modelling the Domain Name System”, she explained that a combination of Microsoft features and misconfigurations essentially caused a slowly paced massive distributed service (DDoS) attack on the root name server system. What is interesting about this case, is that if the number of machines had been limited to reasonably small number (hundres, thousands, or perhaps even tens of thousands), the problem would not have occurred. However, since the set of machines is so truly enormous, the problems do occur.

It is well-documented that humans find it hard or even impossible to apply their mental models to extremely large systems. As a consequence, problems occur constantly, when developers build systems that do not scale. Often, the results are only detrimental to the application itself, but occasionally the problems have knock-on effects in other areas (see also Section 6.2).

Another example in the security area will clarify this further. The Welchia worm, also known as the “Nachia worm”, is a computer worm first discovered in 2003. It exploits a vulnerability in the Microsoft Remote procedure call (RPC). It is designed as a *good* worm, that automatically downloads and installs patches (to stop other worms like Slammer). However, it had problems. First, people did not like the fact that it penetrated and modified their system without consent. Second, and more relevant to this section, is that it was way too successful. It spread so quickly and replicated itself so fast, that it created vast amounts of traffic (due to its transmission method), thereby slowing down the Internet and the Microsoft website.

In the future, we expect recurrent patterns of security vulnerabilities that come with scale. Specifically, in a 100-billion node network, composed of what we consider the traditional Internet, but also smartphones, networked vehicles, and a variety of sensors implanted in our everyday environment, will bring out and transform old vulnerabilities.

Possible solution(s). We believe there are three approaches to counter threats due to scale. First, we must study and understand the interdependencies between systems. Without a clear understanding of possible side effects there is no way to move forward. Second, we should model larger systems in our security evaluations. It is no longer sufficient to assume systems of hundreds or thousands of nodes, we have moved firmly into the domain of tens of millions. Last, whenever possible, we should form clear boundaries between systems. Strict isolation can break the domains in smaller, more manageable sizes.

http://www.caida.org/publications/presentations/2003/nmspi0305/nmspi0305.pdf

Symantec information on Welchia / Nachi http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html

6.4 System maintainability and verifiability

Threats. Smart environments often consist of a large collection of sensors, controls, computing equipment and output devices, as well as connections between them. Consider a smart home. It may contain a multitude of media devices and sensors, heating, lighting, phones, refrigerators, washing machines, blinds, sprinklers, and both electronic (cameras, motion sensors) and physical (locks) security devices. Ideally, it would seem, all these systems should be integrated.

It is unlikely that all these devices are made by the same vendor, so there may not be an obvious entity to go to when there is a problem. Even if they are made by the same vendor, there are many things that can go wrong. The probability of some devices malfunctioning or interacting with other devices in an undesired way is enormous due to the large number of devices (all devices in a house times all the houses).

Once there is a problem in such a smart-environment, it may be incredibly difficult to debug it. In essence, the smart home is a complex distributed environment with many nodes that all interact in unpredictable manners. It may be that a problem with one device (say the locks) is caused by a completely unrelated device (e.g., because a bug somewhere causes a denial of service attack against the first device).

Things become even more complicated when software or hardware is updated, added to, or removed from an existing smart environment. How can we be sure that the complete system still behaves properly? Testing a complex distributed system is exceedingly hard and exhaustive testing is probably impossible. It may well be that nobody really understands the systems and that patching one part creates problems in another part (a fairly common problem with patches).

The real problem is that some errors do not manifest themselves until much later. Moreover they may occur only in rare situations (for instance, due to unusual race conditions, or exceptional circumstances like a very hot summer, or very cold winter). Verification would be a possible solution, but it is hard, given the scale and complexity of current systems.

Parallelism and race conditions only serve to excerbate the problems. Since threats related to parallelism is treated in a separate section (Section), we will not discuss this any further at this point. It is clear that all problems due to parallelism are directly related to maintainability (when you update, will you introduce bad race conditions?) and verifiability (will you be able to verify that the system is race or deadlock free?).

Formal methods and verification may be a possible answer, but unfortunately, they do not scale too well. The current state of the art of formal verification is that researchers in Australia have in a 20 person year effort managed to verify a single, tiny microkernel. It is unlikely that researchers will be able to scale this up to really large systems in the foreseeable future.

Klein, Gerwin; Elphinstone, Kevin; Heiser, Gernot; Andronick, June; Cock, David; Derrin, Philip; Elkaduwe, Dhammika; Engelhardt, Kai et al. (October 2009). “seL4: Formal verification

Moreover, for verification we need formal specifications. It is exceedingly difficult to write such specifications and non-trivial to check whether the system actually complies with the specifications. One of the problems is deciding who will write the specifications? The hardware vendor? The software vendor? The vendor of the overall system? All of them? If so, how accurate is it likely to be, and, again, how will you be able to update the model.

Possible solution(s). Debugging, upgrading, testing and verification, should all feature prominently in the design of a smart environment. They should not be an after-thought. In our opinion, integration should be limited. While communication between some subsystems should be possible, the number of contact points between modules should be limited. Some subsystems may have to remain isolated from the others completely. Centralization of intelligence and standardized interfaces will also help stem the complexity.

On the longer term, we see a demand for a *simple, formally verifiable* language to express behavior in a complex environment. The "smarts" of the smart environment should be expressed in this language. Verifiability should be sufficiently simple to allow verification each time the software and/or hardware configuration changes.

6.5 Hidden functionality

Threats. One threat of paramount importance is that of hidden functionality in systems, and in particular, in software. Hidden functionality may be almost any functionality, but common examples are backdoors, i.e. secret and undocumented entries to a system, and Trojan horses. Such functionality can be introduced into the system by accident, but the most common reason is that somebody, for example, the designer or maintenance engineer, enters this functionality for his own, in many cases malicious, purposes. In other cases, it is introduced for commercial reasons. Regardless of its purpose, the idea is that this extra hidden functionality is not known by the authorized user and the rightful owner of the system.

It is evident that such functionality presents an enormous threat. Not only is it unknown, but it is also put into the system in such a way that it is very hard to find it. Furthermore, this functionality is totally uncontrolled and can lead to a large range of very detrimental impacts on the system.

As an example, in the U.S. the possibility of malicious hardware used for espionage, or even for terrorist activities is considered an emerging threat (as discussed in Section 4.2). Most hardware fabrication is nowadays outsourced. Circuits can be added on chips at the fabrication plant to offer a backdoor to potential attackers, or perform some other action. It is technically very hard for vendors to detect whether the produced hardware follows their design to the letter.

of an OS kernel". 22nd ACM Symposium on Operating System Principles. Big Sky, MT, USA. `http://www.sigops.org/sosp/sosp09/papers/klein-sosp09.pdf.`

Possible solution(s). It is very hard to find solutions to this problem. Any type of remedy would imply that you must be able to prove, or at least make plausible, that no such functionality exists. Unfortunately, there are significant theoretical obstacles in proving the *absence* of something. It is certainly possible to find and remove such functionality, but to verify that there is none left after removal is extremely hard. Still, the only possible solution would be to develop better validation and verification methods and tools. A methodology for measuring security could be one of them as well as runtime detection of any unknown (malicious) functionality.

In the short term, potential solutions to this problem might involve the use of secure and trusted fabs for critical hardware, such as the one used in aviation and the military.

6.6 Threats due to parallelism

Threats. Currently, we are experiencing a revolutionary growth in multi-core and hardware multi-threaded systems. The reason is that hardware architects can better exploit the exponentially increasing density on microchips by increasing the number of (slower) processing units than attempting to increase single-threaded performance. The consequence of this is that we will soon experience a shift in which software systems can achieve performance improvements only through parallel computation. Unfortunately, humans are not very good in managing parallel processes, and software developers are no exception. Also, the tool support for the development of parallel programs is in its infancy. Thus, we can expect a significant increase of bugs and security vulnerabilities due to parallelism, in particular, race conditions.

Race conditions in software systems that may lead to security violations is not new in the security community. Typically, the attack involves some privileged program that examines the state of a resource before proceeding in a critical operation on that resource. The time gap between the check and the actual operation gives a window of opportunity to the attacker to modify some aspect of the target resource for their own benefit. The vulnerability is caused due to the belief of the privileged program that it is the sole executing entity on a system.

Let us also not forget that multi-core processors are not limited to use on personal computers, but are also used on phones, sensors, and so on and so forth. One may argue that personal computers, which run mostly tested operating, should be able to cope with the increased threat level. But, other devices like mobile phones and sensors, utilize simpler, embedded operating systems, perhaps with fewer facilities to combat this type of attacks.

Possible solution(s). We believe there are specific steps we can take to counter threats from the increased number of hardware processors operating in parallel. We should start by reevaluating our applications and operating environments. Applications have been mostly developed with the single threaded model in mind. Even multi-threaded applications have been designed with only a few tens of threads in most cases. Software architects should begin fundamentally changing the way they build software. On the other hand, operating systems must change to handle the more complex, and parallel hardware. Hardware virtualization is not a panacea but it will certainly help in some respects (although virtualization can also lead to problems, which were discussed in more detail in Section 4.3). To assist programmers of both applications and operating systems, we must invent new programming languages that are designed for highly parallel and multiprocessor environments.

Chapter 7

Threat Category: Data Manipulation

7.1 Overview

The data manipulation category covers threats that stem from the fact that people (and systems) store more data online, and this data is becoming increasingly valuable and sensitive. In addition to the problem of storing data that can be stolen, we also consider the problem that data can be fabricated. That is, attackers can create seemingly legitimate input to systems that can cause security problems when these inputs are trusted.

One major reason of the increased value of data that is stored online is that it is related to information that people consider private. Whether this information are personal stories and pictures uploaded to a social network, or sensors that record pictures and sounds from a person's living room. In both cases, users would like to restrict access to this data. Attackers, on the other hand, who see benefit in gaining a hold of this information, develop new techniques to bypass access restrictions.

In the following, we list and discuss the threats in the data manipulation threat category that we have identified.

7.2 Privacy and ubiquitous sensors

Threats. Privacy has been a concern in ICT-environments from the outset. In this section, we identify several interesting aspects to privacy violation, other than well-studied topics such as spyware, faulty encryption, etc.

We are not the first to observe the threats accompanying the introduction of ubiquitous sensors and mobile computing equipment. Security cameras, keycards, parking sensors, and RFID ranks among the conspicuous examples of such sensors, but the list is endless. In the near future, it may be difficult to engage in activities outside the home without leaving a trail of electronic footprints. Such information in the hands of attackers lends itself to abuse.

What is new is that sensors like phones, PDAs, RFID tags, and media players are increasingly with us *always* and *everywhere*. A compromised phone might be used by attackers to obtain audio and video recordings out of classified business meetings, or even our supposedly private bedrooms and bathrooms. In addition to (visible) security cameras, and other devices in the public space, we must now consider our own devices that may have been compromised and betray us. The closest analogy is that of spyware in a traditional PC that tracks our Internet interests, or gains control over the computer's webcam. However, with ultra-portable devices with a plethora of sensors, the scope for "spying" expends tremendously.

Another new development in the realm of privacy, includes RFID tags. RFID tags worn by users in clothing or "smart cards" may be read by readers close to the wearers and identify and store their presence in sensitive areas. Knowledge about where you have been, what you have done, and what your interests are can be exploited in various ways.

Possible solution(s). Beyond securing the devices that may be compromised, it is hard to find an immediate technical solution for these privacy problems. An important partial solution is to educate users about how ultra-portable devices will affect their privacy.

7.3 False sensor data

Threats. If privacy is concerned with the undesired *leakage* of information, the opposite threat is that of data *fabrication* or *falsification*. Suppose sensor data identify a user as having visited an embarrassing or illegal venue (e.g., the red light district, or an illegal neo-nazi rally). Can we be sure that the user really was present? To what extent can we trust sensor data connected to the future Internet? Unfortunately, the answer cannot be unequivocally yes due to two reasons: (a) potential bugs in the devices, and (b) potential data manipulation by attackers.

Buggy devices yield wrong information. A common example of a buggy sensor is the barcode reader in the supermarket that double-charges a product. Similarly, supposedly-disabled RFID tags on clothing or other products frequently set off alarms when a client exits the shop, often creating embarrassment. It has happened to most of us, and these are the simplest examples of what we call buggy devices. While barcode readers result in modest overcharging, other types of buggy sensors are more serious. In some places, automated parking sensors are used to identify your car, and cars are tracked on motorways to pay as-you-go. In several countries, public transport is accessed using a smart card or phone, directly linked to the user. As the number of sensors increases, so does the probability of false reports.

Similarly, as Internet Protocol (IP) telephony becomes more popular, buggy telephones (or buggy call protocols) may misbehave and dial wrong numbers. Worse, they can be manipulated to dial numbers that are embarrassing to the user. For instance, call records could indicate that employees frequently converse with the competitor (or more embarrassing, perhaps, paid sex lines). It is unclear whether an employee will be able to convince their employers that something went wrong.

Perhaps the most significant threat is that phones, sensors, or databases can be hacked, and data can be falsified. Most information is stored in centrally-controlled databases. Take, for example, your telephone company. Whether you are at home or roaming away from it, all your phone calls are logged by your provider in a huge, centralized repository. A sophisticated hacker or a person with knowledge of a company's internals can alter all the information about your phone calls. Such actions can have significant repercussions in one's life. For example, one may be framed by being linked via telephony records to criminals. Mobile tracking, shopping activities, car parking, they all can be manipulated to create a virtual clone of yourself with unknown implications. In the digital world, where everything is connected via the Internet, planting of "evidence" is both easier to do and harder to detect.

Besides data fabrication, attackers of course may also remove traces. The motivation for doing so may vary. A recurring example in movies and books is that in which potential alibis are deleted, but it is not hard to imagine other uses.

Falsifying sensor data is simply a new variant of form of traditional data tampering. Tampering with data for one's own benefit is hardly new. For instance, in the 1983 Hollywood production "War Games," Matthew Broderick was shown tampering with the school database to change his grade from a fail to a better grade. However, tampering and fabrication to hurt someone is less common, but should be taken more seriously in a world where information is stored in many places.

We have sensors and databases in hospitals, in banks, in organizations, in police and justice departments. Given the proper motive, a person with access to such information can easily incriminate someone, or worse. Consider a determined attacker who tampers with a patient's medical records. When the patient receives treatment, the doctor consults the e-record to check the patient's medical history. The original medical record indicates a blood allergy to specific drugs. An altered medical record hides this information. Wrong treatment to the patient may have dangerous consequences. As another example, attackers wanting to hurt or embarrass someone may "invent" a serious medical condition, a poor credit history, or a criminal past, all of which may reduce his/her eligibility to certain programs, jobs, or opportunities.

Possible solution(s). It is unlikely that we will be able to prevent falsification of sensor data altogether by means of technical solutions. As a result, the reliability of electronic data should have legal implications. Clearly, the (un-)trustworthiness of sensor data creates a legal void that needs to be filled.

Most easy solutions are wrong. Admitting sensor data as reliable proof makes little sense if the data may be unreliable, and especially if the data can be altered. But clearly, there is a link between the sensor data and reality in most cases, so we cannot altogether dismiss sensor-based evidence either. Current legislation already looks at some of these issues (e.g., the legal status of footage from a surveillance camera), but the scale at which a multitude of sensors will track persons and objects in the future is such that re-thinking legal implications is important.

The issues that need to be taken into account ranges from evidence based on individual sensors to collections of sensors, and incorporates both agreements between sensors and anomalies. Ideally, there should be a way of establishing the reliability of sensor data. This is not easy. For instance, we cannot say that a certain sensor has a reliability of 98.5%. However, we may be able to categorize the security of devices. Arguably the most important question that needs to be answered is how people who are accused on the basis of sensor data can defend themselves.

7.4 Threats related to social networks

Threats. A social network is a social structure that is made up of nodes that represent individuals or organizations. These nodes may be tied to each other by properties such as friendships and general interests. Recently, the popularity of social networks that might focus either on business-relationship or friendship has dramatically increased. As social networking sites such as XING [158], LinkedIn [85], Facebook [33], StudiVZ [141] and MeinVZ [94] have been gaining popularity among Internet users, miscreants started to abuse most widely-used social networking sites for their nefarious purposes.

The nature of information social networking users provide by registering to the network is sensitive and attractive. Typically, users enter their real e-mail addresses and provide information on their education, friends, professional background, interests, sexual preferences and activities they are involved in. Hence, from the attacker's point of view, access to this type of detailed, personal information would be ideal for launching targeted, social engineering attacks, now often referred to as spear phishing [59, 140]. Furthermore, the collected e-mail addresses and personal information would be invaluable for spammers as they would have access to e-mail addresses that belong to real people and have information about the people using these e-mail addresses allowing them to efficiently personalize their marketing activities, tailored according to the knowledge from the target's profile.

The relation between users on the current popular social networking sites is based on strong trust. The malware authors may leverage that fact and abuse the service by using the environment as their infection medium. Obviously, if an attacker manages to build this relation between the victim, she can easily deceive him to install her malware since the victim will think the attacker is a trustworthy friend. To this end, the attacker may choose to perform impersonation attacks, which are proven to be doable not only by researchers [101] but also real-world attacks that are observed in the wild [34].

The attacks performed in the past show that the criminals started to focus on popular social networking sites such as MySpace, Facebook, Orkut and so forth.

They were mainly trying to deceive users to install their malware by directing them to fake codec sites or attracting them to install some social networking site specific applications. The latest, and also the most interesting attack, was seen in February 2009, and it was carried out by Koobface worm's latest version [73]. The malware installed after infection of the victim was trying to steal credentials of various social networking sites. The fast evolution of the type of the attacks targeting social networking sites can be interpreted as more serious attacks can appear in the nearest future.

The more compromised users an attacker has, the more massive attacks she is able to perform. Obviously, if a feasible way to launch the social-networking attacks in an automated fashion can be found, the attacker may easily and quickly achieve her goals. To prevent attackers from automatically accessing and abusing their services, social networking sites make use of CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). A CAPTCHA is a type of challenge-response test that is commonly used to determine whether the user of a certain application is a human being or a program. The key feature of CAPTCHA algorithms is the ability to generate tests that are at the same time easily solvable by humans, but very difficult to solve for a computer application. Clearly, using CAPTCHAs that are very difficult to be broken by programs most likely can stop emerging social networking attacks from the beginning. However, most of the popular social networking sites do not put enough effort into making automated crawling and access difficult.

Unfortunately, even if the social networking sites use CAPTCHAs that are very difficult to be solved by automated programs (e.g., reCAPTCHA [153]), the criminals who own a botnet with numerous bots may evade the CAPTCHA obstacle. In the beginning of 2009, the spammers used a botnet to crack Microsoft Live Hotmail CAPTCHAs to create a large amount of accounts. Similar technique can be applied to social networking sites for other malicious activities as well.

Possible solution(s). A prerequisite for being able to access personal information in a social networking site is a confirmed personal relationship with the person who is concerned. Once the attacker establishes such a connection, which is based on strong trust, he can insidiously make the user perform various attacks without her being aware of the involvement in the attack. The most challenging task for the attacker seems to be persuading the victim that a friendship request is coming from a real friend. However, recent research showed that it is not as difficult as it is thought to fool social networking users. Obviously, the user is the weakest link in social networking sites. Many are not security-aware, and there is much implicit trust. One solution that could improve the security of contact requests would be to provide more information to the receiver on the authenticity of requests.

Since it is very difficult to make all of the users be aware of authenticity and privacy issues on social networking sites, the biggest part of the work on making social networking sites more secure has to be done by the social networking providers. Recent research showed that not all of the social networking sites track anomalous behaviour such as crawling, consecutive CAPTCHA solving attempts, large number of similar activities done by one account, etc. Hence, if the service provider applies anomaly detection techniques, they may stop or at least slow down propagation of the malicious activities on social networks.

7.5 Online games

Threats. Online gaming has become a billion dollar industry, and game companies are making significant revenues from subscription charges. Of course, whenever there is money involved, it attracts the interest of cyber criminals who look for easy prey and quick profits.

A trend in online games is that resources, such as weapons and real estate, have a real-world monetary value. In some games, users buy the virtual money in the currency used in the game with real world money and use that virtual money to buy objects. In other case, users trade among themselves in online markets that are not connected to the game. In both cases, the theft or destruction of a user's objects represents a real loss.

A related issue is that users can pay for good and services in a game using credit cards. Compromising the game may put the user at a risk of online theft of his credit card data.

Finally, a significant problem in online games is that every player has a game client (program) that is locally running on his or her machine. This game client often plays an important role in verifying user inputs and making sure that the player's actions are conforming with the game world. Unfortunately, attackers have always found ways to bypass client-side protections. Hence, we can expect to see a variety of attacks against the game servers, which are supposed to be prevented by proper game clients.

Possible solution(s). Online gaming is a market that is worth billions of Euros. Interestingly, however, there has not been much systems security research for protecting online games. We believe that as the online gaming market grows, we will start to see more attacks in this area.

We believe that research is needed in investigating the security of online games and virtual worlds. One possibility is to look for techniques for eliminating or reducing the need to have state on the client. This would reduce the ways in which attackers can tamper with the game client to perform unintended interactions with the game world.

Chapter 8

Threat Category: Attack Infrastructure

8.1 Overview

As the popularity of the Internet increases, so does the number of miscreants who abuse the net for their nefarious purposes. A popular tool of choice for criminals today are *bots*. A bot is a type of malware that is written with the intent of compromising and taking control of hosts on the Internet. It is typically installed on the victim's computer by either exploiting a software vulnerability in the web browser or the operating system, or by using social engineering techniques to trick the victim into installing the bot herself. Compared to other types of malware, the distinguishing characteristic of a bot is its ability to establish a command and control (C&C) channel that allows an attacker to remotely control or update a compromised machine. A number of bot-infected machines that are combined under the control of a single, malicious entity (called the *botmaster*) are referred to as a *botnet*. Such botnets are often abused as platforms to launch denial of service attacks, to send spam mails, or to host scam pages.

Several studies show parallels and draw connections between malicious Internet activity and the underground economy. For example, Provos et al. provide technical details on how cyber-criminals use web-based malware to their advantage [126]. The aspect of an underground economy that is fuelled by financially motivated cyber-criminals is highlighted by Franklin et al. [41]. In a recent paper, Holz et al. study the structure and profits of keyloggers [50].

The attack infrastructures threat category covers threats that are related to the fact that adversaries actively develop and deploy offensive platforms (such as botnets). That is, adversaries no longer perform hit-and-run attacks, but they establish operational bases on the Internet used to carry out malicious campaigns.

8.2 Underground economy support structures

Threats. Over the last few years, there has been a dramatic change in the goals and modes of operation of malicious hackers. As hackers realized the potential monetary gains associated with Internet fraud, there has been a shift from "hacking for fun" (or bragging rights and celebrity within and outside the hacker community) to "hacking for profit" [41, 148]. This shift has been leveraged and supported by more traditional crime organizations, which eventually realized the potential of the Internet for their endeavors.

The integration of sophisticated computer attacks with well-established fraud mechanisms devised by organized crime has resulted in an underground economy that trades compromised hosts, personal information, and services in a way similar to other legitimate economies [74, 136, 137]. This expanding underground economy makes it possible to significantly increase the scale of the frauds carried out on the Internet and allows criminals to reach millions of potential victims. Also, criminals are taking full advantage of sophisticated mechanisms, such as the service bots used on IRC channels to automatically verify stolen credit card numbers or the use of fast-flux networks [51] to create attack-resilient services.

The emerge of the underground economy has resulted in the existence of well-funded adversaries that have the incentive and the means to create better, stealthier malware. In addition, it has also led to the development of novel services that cater to the needs of the underground economy. For example, malware that steals sensitive information requires a way to leak this data back to the malware author (or controller). Often, compromised machines (so-called drop zones) are abused for this purpose [49]. Another example are trading forums (such as IRC channels, web site) that are leveraged by criminals to exchange stolen information for money [41]. Finally, there are services needed to launder or to exchange money. Recently, the use of e-casinos has become a popular means to transfer money from one party to another. To move money from party A to B, both join the same game (table) in an e-casino. Then, the player(s) controlled by A deliberately play weak and lose their bets to the player(s) controlled by B. Such transactions appear legitimate, and they are difficult to identify as illegal money transactions. The different services and novel schemes that are forming in the wake of the underground economy are of significant concern. In particular, we require actions to counter and disrupt these services and transactions to attack the underlying platform on which criminal activity thrives.

In addition to the novel services and platforms used by cyber-criminals, it is also possible (and likely) that they seek novel business opportunities. One such business model leverages the fact that a botmaster controls a large number of desktop machines that store a significant amount of possibly valuable information [27]. While it is easy (and already practice) to search this data for financial credentials, it is also possible to monetize other, more specific information. For example, one of the compromised machines might contain Word documents about a certain company that is interesting to and relevant for a competitor. To connect the data with potential buyers, the botmaster could decide to rent his botnet to customers that are then allowed to perform a number of desktop searches on the compromised hosts. Thus, the botmaster does not require to know in advance what information is valuable. Instead, he just sells access to his data to criminals that are more specialized in looking for certain kinds of data [54].

Possible solution(s). One reason why the underground economy is flourishing is because it is very difficult to trace back the attackers [54]. Also, the cost of running illegal operations for the attackers is low. It is clear that it is important to disrupt the underground economy, possibly by using offensive techniques to increase the cost for the attackers. For example, a possible defensive solution could be to inject a large volume of false information as a response to the attacks launched by the attackers [41]. As a result, the attackers would face the problem of identifying which stolen data is valid and which data is fake, and the cost and the difficulty of the attack would increase.

Another possible offensive, perhaps controversial, defense technique could be to automatically launch DoS attacks against illegal web sites that are operated by the attackers. By making these sites inaccessible, potential victims could be prevented.

8.3 Advanced malware

Threats. A popular tool of choice for criminals today are so-called bots. A bot is a type of malware that is written with the intent of compromising and taking control over hosts on the Internet. It is typically installed on the victim's computer by either exploiting a software vulnerability in the web browser or the operating system, or by using social engineering techniques to trick the victim into installing the bot herself. Compared to other types of malware, the distinguishing characteristic of a bot is its ability to establish a command and control (C&C) channel that allows an attacker to remotely control or update a compromised machine. A number of bot-infected machines that are combined under the control of a single, malicious entity (called the botmaster) are referred to as a botnet. Such botnets are often abused as platforms to launch denial of service attacks [97], to send spam mails [63, 128], or to host scam pages [8].

Most currently active botnets' C&C mechanism is based upon the Internet relay chat (IRC) protocol. There are various reasons for the popularity of IRC among botmasters: IRC enables small and simple client implementations in the bot software, it allows the botmaster to use off-the-shelf clients for commanding his bot army, and it allows to use publicly available, legitimate servers for hosting the C&C rendez-vous point, while at the same time offering built-in functionality for access control, to keep out security researchers, or other botmasters trying to inflict harm on their rivals. However, most importantly, IRC has been chosen as the C&C protocol in a few original bot implementations, from which an overwhelming fraction of today's active variants are still derived. A brief overview of the most important variants of IRC-based bots as well as their functionality can be found in [13].

While IRC served botmasters well in the past, the anti-malware industry, security researchers, as well as network administrators have taken advantage of several shortcomings IRC exhibits as a C&C protocol. First, compared to other application layer protocols, such as HTTP, IRC is not a main-stream protocol used by a great number of people for serious purposes. For many, IRC has even turned into a synonym for botnet C&C. Many firewalls, especially in company networks, filter IRC traffic, an thus, render any successfully compromised machines useless for the botmaster. Second, for most of the popular variants there are network signatures that identify infections when deployed in a network intrusion detection system. Owners of publicly accessible IRC networks pay attention to identify C&C channels on their servers and take them down. Third, and most importantly, the C&C structure of an IRC botnet exhibits a single point of failure: The IRC service. The botnet is not robust to failures, caused either by technical malfunctions, or by intervention of anti-malware institutions aiming to shatter the botnet. By taking down the C&C channel, the botnet is irreversibly destroyed. For these reasons, a recent trend is that IRC is no longer considered a safe and efficient means of communication for botmasters.

While script-kiddies might continue to use IRC, and thus, the majority of botnets will likely still use it in the near future, more professional attackers have begun to explore alternative means of enabling C&C communication. These miscreants are motivated by an outlook for huge financial profit, and do not refrain from investing money and time into developing custom-tailored solutions that remove some, or all of the drawbacks that IRC brings. Recently, numerous botnets using alternative C&C protocols have been detected and monitored by the anti-malware community. The two most wide-spread alternatives to IRC as a C&C medium are HTTP, and peer-to-peer protocols, such as Overnet.

The utilization of HTTP as C&C protocol, above all, camouflages the C&C communication within a large amount of traffic transported with the most commonly used application-layer protocol existent on the Internet. Unfortunately, it is not possible to detect suspicious signs towards the presence of a bot infection in the network traffic, unless costly deep packet inspection is performed on all web traffic. Firewalls do not pose a problem, since Web traffic usually must be allowed to trespass to fulfill usability requirements. Even though HTTP botnets can be taken down by either disabling the server itself, or preventing the resolution of the domain name, bots can iterate over a list of registered domain names in order to locate a server that is active. These domain names can even be computed dynamically, based on the current time. So, even after having lost the botnet, botmasters can eventually regain control. One of the most well-known bot implementations using HTTP is the infamous Conficker worm. More detailed information about Conficker is presented in [81].

Other botnets, such as the Storm worm, make use of peer-to-peer networks to communicate the botmaster's commands to the bots. By using well-established file sharing networks, such as Overnet, the botnet immediately takes advantage of a high number of peers to use as communication partners. Also, its traffic remains completely unobtrusive in front of the ubiquitous background of benign file sharing traffic.

In [44], an overview on the field of peer-to-peer botnets as well as the Storm worm is presented. A more detailed discussion on the Storm worm, including static analysis results of the worm binaries, can be found in [125]. The authors also explain how Storm leverages the widely-used Overnet peer-to-peer protocol to put in place its C&C network. In [52], the authors elaborate on tracking and analyzing peer-to-peer botnets, and demonstrate their strategies on the Storm worm. In addition, approaches for infiltration and mitigation are developed.

In addition to switching to more powerful, and less suspicious protocols, attackers make use of simple, yet powerful tools to avoid detection and retain secrecy about their intentions. By transmitting the commands in an obfuscated form, instead of in the clear-text, botmasters now successfully avoid detection by current automatic (intrusion detection) systems to identify bot infections. By using classic cryptographic methods, they can even thwart manual efforts from malware experts to decipher the commands, unless an instance of the bot software is obtained and reverse engineered.

Threats imposed by botnets will continue to increase. In the near future, botnets will be less-dependent on IRC but more on protocols such as P2P, and even instant messaging infrastructures.

Possible solution(s). While throughout the past years many efforts have been made to mitigate the botnet threat, these efforts have only been partially successful. The changes in the tools and tactics used by botmasters and malware authors clearly call for further research in order to identify, correctly specify, and filter botnet traffic. Due to the ever-increasing presence of botnets on the Internet, there is a need for automated systems aiding researchers in their work. Because of the rising diversity in the C&C structures of current botnets, completely new methods for deriving and implementing means of detection must be developed.

Chapter 9

Threat Category: Human Factors

9.1 Overview

In several working groups, the human was identified as the sometimes weakest link in the system. This is especially true in critical systems, where the *human – system* interaction affects safety and any failure could have serious consequences.

There are several issues related to this category. For example, *complexity* (discussed in Section 6) makes it difficult for the human operator to understand the complete system and thus adding to the risk that the wrong decision will be taken in a crisis. However, the same system complexity also makes it difficult to build systems that can run autonomously without failures, thus reinforcing the need for a flexible human operator that can take care of special cases that the programmers of the system never foresaw. Thus, there is a need for a clear communication channel between the system and the human operator, i.e., the user interface must be well developed.

In this section, we consider the following threats that have a clear human aspect to them.

- *User Interfaces* are important so that current system status can be clearly communicated to the human operator, including any ramifications of choices made.
- *The Insider* have extensive knowledge of the system and may misuse the system for his own gains.
- *Safety takes priority over security* in many critical systems, even though these two concepts are interrelated.
- *New vectors to reach victims* have been developed lately and affect the normal user. Finally,
- *Targeted attacks, spear phishing* is also an attack form that is more common and thus need special attention.

9.2 User interface

Threats. The human plays various roles in systems, be it an industrial type of system or just for the home application. In the former, the roles include being operators in control rooms, engineers taking technical decisions, managers and decision-makers for future strategy development. It has been estimated that in some situations, human reliability falls from 10^{-4} to 10^{-3}, whereas a system's reliability is 10^{-9}. There are incorrect interactions with the system, other operator errors, and interdependencies. The human being is a serious factor when considering overall system security.

There is a lack of understanding of the overall (critical) system, since its complexity is continuously increasing. This is growing to become a serious problem. Large networks are hard to encompass and their comprehension goes beyond the capacity of the human brain.

Similar discussions can be held for the latter category described above, i.e., the normal user. A lot of today's threats like cross-site scripting, phishing or similar attacks could be mitigated by various techniques. In reality, however, these solutions often require a user to have at least a certain understanding of what the threat means to him. Furthermore, this knowledge is important to let a user decide what a specific warning dialog means. Even today, users become "resistant" to dialogues. They strongly tend to get rid of annoying interruptions by clicking OK on each appearing question. This problem is not a technical one, but of the user interface. Nevertheless, it is imperative to wrap new solutions to upcoming and even existing threats in understandable and discreet user interfaces to make sure, they are properly used. This overload is a constant problem that is very likely to persist for a long time and hinder solutions for security problems to catch, even if they already exist.

Possible solution(s). The education and training of personnel working in critical systems is a constant task that can help maintain an up-to-date knowledge on systems and networks. The awareness of security risks should be raised. There are many bad practices (e.g., running unpatched versions of software, using default configurations and passwords, etc.) that could easily be removed by making people understand the role of security measures. A sound and evolving security policy in the organizations is needed to mitigate security risks. There are approaches to model the user (*cognitive modeling*) and user-interactive properties that could be used to improve the interaction of the users with the systems.

Another approach is to model and design the systems in such a way that they are more easily comprehended and understood. This would include e.g., structural design, encapsulation, intuitive interaction interfaces, etc.

9.3 The insider threat

Threats. Insiders are employees with experience of and knowledge about a system. The threat from the insider lies in the risk that a trusted employee betrays their employer by conducting some kind of malicious activity. Insider betrayals comprise a broad range of actions, from theft or subtle forms of sabotage to more aggressive and overt forms of vengeance, sabotage, and even work place violence. Insider activities cause financial losses to organizations, have negative impacts on their business operations and damage their reputation. It is of particular concern to the financial sector where the problem is known, but also other sectors are realizing the damaging effect an insider can have.

In [114], it is argued that the nature and seriousness of the threat requires a combined view of physical and IT security systems and policies. Although physical and cyber threats from insiders manifest differently, the concepts are quickly converging as many potential attacks bear characteristics of both physical and IT sabotage, fraud, or theft.

The “insider threat” to critical infrastructure is defined in [114] as:

> one or more individuals with the access and/or inside knowledge of a company, organization, or enterprise that would allow them to exploit the vulnerabilities of that entity’s security, systems, services, products, or facilities with the intent to cause harm.

One of the main findings of that particular study is that any modeling of the insider threat needs to take into account the potential of combined physical – cyber attacks. Moreover, a coordinated attack combining an insider attack with an external attack could have multiplier effects and could be much more destructive than a simple one-dimensional attack.

Some interesting results from a study on the insider threat [69] show that a negative work-related event is most likely the trigger to most insiders’ attacks. Furthermore, the majority of insiders planned their activities in advance. An observation is that the majority of insiders were granted system administrator or privileged access when they started work, although less than half of the insiders had authorized access at the time of the incident. An interesting point is that both unsophisticated and relatively sophisticated methods for exploiting system’s vulnerabilities were used. Remote access was used to carry out the majority of the attacks. Many times, the insider attacks were only detected when there was a noticeable irregularity in the information system or when a system became unavailable.

Possible solution(s). Effective strategies for discovering an “insider” is an open research question. The recommendations from [114] include low-cost, easily implemented policy solutions for near-term effect: education and awareness, employee screening, technology policy, information sharing. In the long-term aspect, further guidance, findings, samples, and tools are needed. Some solutions for IT systems/cyber security could be the following: to use integrated IT and physical security system tools to identify rule violation patterns for potential insider threat behavior; to use dual protection access technologies (e.g., biometric, key card or encryption key verification); to use dual control access mechanisms to protect high-value systems and processes; to manage access, integrity and availability of computer systems (e.g., identity management system). Control over creation and termination of user and administrator accounts and maintaining security/access rights should be done by segregation of duties. Using data loss prevention tools could help stopping the leakage of information outside the network and can be a measure to detect an insider activity.

9.4 Safety takes priority over security

Threats. In the domain of critical systems both safety and security are important but in certain scenarios, safety takes priority.

Based on the tradition of safety-critical systems, safety is and has been emphasized over security. Examples exist in the industry corroborating this statement. For example, passwords are sometimes avoided by intent. It is reasoned that sometimes it is very important to immediately be able to control a process (to stop it from reaching a critical point), and a password would only slow down the operators. Thus, no regards to integrity or access control exists in such a system and such features cannot easily be added later, or added to one part of the system if another part lacks such support.

Giving priority to safety is not just a traditional vision. It is justified by the potential losses after a safety incident. The safety of critical systems (CS) is important because of CS interaction with the physical world and the possible risks of that interaction. Security is usually considered being of less importance compared to the major safety issues of the actual CS. With the extensive use of ICT in critical systems, however, security should be considered more seriously, since security and safety are very interrelated. Problems with security can lead to safety issues. Thus, a security attack can lead to a safety problem and endanger lives.

Complicating the issue is the fact that control system professionals are often not aware of security risks, since these are not considered part of the normal system operation. The emphasis in control systems is on safety and availability aspects. On the other hand, IT security specialists use known techniques from a normal ICT system to introduce security, but may be missing important safety and control characteristics of the specific CS. Traditional security measures are usually not directly applicable to critical systems. Delays that may be caused by the operation of security tools are not acceptable within such systems. Critical systems, especially the ones with a real-time requirement, need to be available around the clock. They cannot be interrupted or restarted to introduce software patches or implement a security mechanism. All security measures should be tested before being implemented to ensure that they do not conflict with control operations.

This lack of mutual understanding between the control and security communities makes the overlooking of security a problem. Control specialists and even the management personnel of organizations are security-unaware and tend to neglect security measures and tools. Sometimes people with little experience or with different primary tasks operate the supporting IT system and they are more prone to do mistakes or ignore security alerts. All these problems stem from the vision that safety is the main priority and security is only a complementary measure to maintain the ICT supporting network properly operational.

Possible solution(s). As we stated previously, the understanding that safety and security are interrelated is of very high importance and will lead to improvement in overall security and safety policy. A better understanding of the domain for the IT security experts is necessary. On the other hand, the control community should be aware of the important role of security measures to safety. Work should be done on changing the mindset. Some simple measures could be to document changes done to the system in order to facilitate the implementation of security tools where they are most needed; keep the control traffic off the business network; document all the software installed on the network, etc. Security should be tailored to the specific characteristics of the critical system. Some new solutions that respect control process requirements for timing and performance might need to be developed.

9.5 New vectors to reach victims

Threats. In the past years, cyber criminals have constantly improved and extended their malicious operations on the Internet. Unfortunately, threats such as worms, viruses, credit card and identity theft, phishing websites and other fraudulent online activities are still on the rise.

Cyber criminals have traditionally employed a number of techniques to find potential victims. The vectors used for reaching victims include mass (spam) email, fake web sites, social engineering, online advertisements served to benign web sites and other forms of online or offline communication. Occasionally, even real-world, hardcopy mail is used as a part of online fraud. For example, a letter might lure a victim to a fake web site and try to convince him to enter valuable, sensitive information. Another interesting example of merging the physical and virtual world has been reported by the SANS institute in February 2009 [131]. In this case, the criminals were using windshield fliers and fake parking tickets containing a link to a malicious web site. By visiting the link, the users were asked to download and install an application to be able to view the pictures of their vehicle.

For cyber criminals, a “victim” for their malicious activity can be either an unsuspecting human or a vulnerable computer system. Typically, the criminals are motivated by monetary profit. This profit can be directly related to their victims. For example, credit card information stolen via a phishing site can be used to withdraw money from the victim’s bank account. On the other hand, a victim could indirectly be part of malicious activities. For example, a worm could turn the victim's computer into a malicious bot, which in turn might be used by the cyber criminal to conduct illegal activities.

Spam remains very popular for cyber criminals, as it has the potential to reach a large number of victims while typically exhibiting low cost. The Spamhaus Project now estimates that about 90% of the incoming e-mail traffic in North America, Europe and Australasia is spam. To evade spam filters and detection, there has been a number of improvements (from the attacker's point of view) [66]. For example, bodies of non-spam text are often inserted into spam emails to defeat statistical detection algorithms, or images containing a spam message are attached to an otherwise innocuous email.

Alternatively, cyber criminals are leveraging other communication media other than e-mail for spam. For instance, instant messaging (SPIM) and Internet telephony (SPIT) are being increasingly used by criminals to send spam or infect their targets [88, 47]. Recently, social networking websites (e.g., Myspace [105] and Facebook [33]), virtual environments (e.g., Second Life [134], Playstation Home [124]) and online games have become attractive targets for spammers, as a large number of users can be reached via these vectors.

In particular, social networks are very appealing for cyber criminals. They can easily create fake profiles and, using the internal search tools provided by the social network, they can identify their victims based on demographic segments or geographical location. To get access to private information only disclosed to the contact's friends, attackers can steal real user identities by creating profiles of real people [80] or by duplicating existing profiles in a different social networks [19]. The information collected in this way can then be used to create targeted attacks (usually called spear phishing [59, 140]). Finally, also more traditional malware infections are moving to social networks, as proven by new worm infections observed in the wild that specifically target MySpace and Facebook users [106]

Another vector that is increasingly used to find victims are compromised, but legitimate web servers. Cyber criminals are injecting malicious code into hacked web servers, which then triggers malicious activity on the visitors' computers. Even security aware users are easily susceptible to this kind of attack, as it abuses the trust relation between the user and the legitimate web site.

The working groupis expecting alternative vectors to reach victims to increase in the future. New technologies and service ideas are constantly emerging and are expected to be quickly exploited by criminals looking for new spreading mechanism and new, more effective way to identify and reach their targets.

Possible solution(s). Existing countermeasures have to be adopted and extended to defend against these new threats, as online criminals are quickly adopting to new trends in technology and user behavior.

Finding victims is one of the first steps a cyber criminal has to undertake. The security community has identified this prerequisite for illegal operations and successfully devised a number of counter-measures. For example, spam and phishing filters are now commonly used by service providers, and anti-virus software and firewalls are easily available for most computer systems. While these systems are far from perfect, they have reduced the effectiveness of some of the malicious activities. Unfortunately, cyber criminals have reacted to these efforts, and are now extending their operations to different vectors to find victims.

New techniques have been proposed to detect IM-based spam [84] and different security companies have started integrating spam blocker functionalities into their instant messaging management systems to filter the traffic and protect the users from unsolicited messages.

There have also been research initiatives to mitigate attacks on social networking sites, both in the direction of protecting the network from the creation of a large number of fake profiles [159], and in the attempt to better protect the online privacy of users' data [45].

9.6 Targeted attacks, spear phishing

Threats. Most attacks on the Internet are aimed at a large number of users. Considering phishing attacks, which are performed to acquire sensitive information such as user names and passwords. It is obvious that the effectiveness of the attack increases if the attackers manage to reach a large number of users. Phishing attacks also require little or no knowledge about the fraud victim, and a single version of a mail text is sufficient to perform the attack. This phishing mail is sent to as many people as possible, hoping that some of them will be lured into clicking on embedded links.

A recent trend is that today, not all attacks are aimed at large groups of people. In recent years, a new kind of attack emerged: the targeted (or spear-phishing) attack. This attack does not target an unspecified group of users, but only a selected group of people or even individuals. Usually, these people are part of a certain community or organization (e.g., employees of a company, a CEO, etc.). Attackers have been attempting to gain access to computers inside organizations in order to steal valuable information such as business secrets. These secrets can be worth significant amounts of money [35].

Whenever a so-called targeted, spear-phishing attack is launched, the e-mail sender information has been faked or "spoofed." Whereas traditional phishing scams are designed to steal information from individuals, spear phishing scams mainly work to gain access to a company's entire computer system. Spear phishing also describes scams that target people who use a certain product or Web site. Scam artists use any information they can to personalize a phishing scam to as specific a group as possible [95].

The number of victims in targeted attacks is usually limited. This has two main reasons. First, if a zero-day exploit is used, targeting individuals is more advantageous as there is a low risk that there will be anti-virus signatures soon that will mitigate the attack. Second, the attack and the attacker will be less likely to be detected. Furthermore, if only individuals are targeted, it is easier for the attacker to cover her tracks.

For 2008, only 0.4% of all spam e-mails were targeted attacks. However, this is a four-fold increase over the previous year, and these attacks tend to cause significantly more damage and have a higher success rate than untargeted phishing. Scammers also take advantage of e-mail reputation hijacking facilitated through the repeated breaking of CAPTCHA schemes employed by major web mail providers. A low volume scam attack sent from a trusted source (for instance, the mail server of a well-known web mail provider) is more likely to pass spam filters and go unnoticed than mass e-mails sent through a botnet [24].

A recent study has shown that the main hurdle to a successful attack is convincing the user to click the malicious link, as users who do so are very likely to divulge sensitive information on the web page they are directed to [77]. More targeted information in scam e-mails raises the probability of users following the link. For example, a user is more likely to follow an alleged link to the website of a bank she is a customer of than to a bank that she is not familiar with. Browser cache sniffing [60] can be used to reveal sites the user has visited to obtain this information.

We believe that there is large potential for more sophisticated targeted attacks in the future. As phishing becomes common knowledge among users, the attackers will shift their attention to targeting individuals and using knowledge about these users that they have acquired on the Internet. The plethora of social networking sites like Facebook, MySpace and Twitter, that have cropped up in recent years, makes the gathering of personal information from the Internet very easy for determined miscreants. These sites also create a new attack vector for so-called “Nigerian” (advance fee) scams, where a hijacked account and the personal information within is used by the attacker to impersonate the victim and ask friends for money [139]. Even when noticed by the account owner, these attacks are hard to stop [109]. A study carried out at the University of Indiana shows that phishing victims are four times more likely to fall for the scam if they are solicited by someone appearing to be a known acquaintance [58].

Possible solution(s). Targeted attacks pose a great threat to organizations and companies. Employees have to be made aware of the fact that even e-mails that appear to come from a legitimate source such as a colleague or a boss can be a fraud. Hence, training and raising awareness is the key to solving the spear-phishing problem. First studies evaluating the efficacy of anti-phishing training programs show promising results. Training against general phishing threats also raises awareness of targeted attacks, and employees working together benefit already from having only few of them trained against phishing [77]. Furthermore, research in the areas of content analysis would be useful in detecting malicious e-mails.

Chapter 10

Threat Category: Insufficient Security Requirements

10.1 Overview

Information and Communication Technologies (ICT) proliferate in many new areas to improve functionality, spreading of information, openness, and reachability. They are used to enrich the services that systems provide, to make them easily manageable and observable, and to connect to other systems and networks. The new areas of application of ICT have often specific requirements and are seldom prepared to introduce these new technologies without any changes and modifications. To add to the natural problems in implementing ICT, come the security challenges. Many of these systems are safety-critical or constitute a part of a critical infrastructure. In some cases, they are critical infrastructures themselves. Their major concern is safety, security often comes as an afterthought. The implemented technologies were not designed to be secure but they have to work in an environment where security is crucial. This section describes threats coming from insufficient security requirements. Putting legacy systems and Next Generation Networks (NGNs) in the same group may seem unusual but with respect to security they face similar problems. Legacy systems need to employ security tools and mechanisms but suffer from limited resources and time constraints. NGNs, on the other hand, seem to have unlimited resources but still need protection means, since they are part of the telecommunications infrastructure but their security is not part of their design. The extensive use of COTS (Commercial Off-The-Shelf) components and systems in critical applications is also considered a threat due to insufficient security requirements because these are general-purpose hardware or software products applied without built-in security.

Some of the common security problems that all these different systems and networks encounter are:

- Implementing ICT that is not meant for security

- Difficult to be tested together before implementing
- Security is added after design
- Since they were not designed for security, these systems bring security vulnerabilities when connected together
- Working with uninterruptable processes

10.2 Retrofitting security to legacy systems

Threats. Security can seldom be retrofitted to an existing system, but economical constraints might still make this necessary. Most critical systems are created to provide a certain functionality. Safety and control characteristics are the natural focus of such systems. Thus, applying security measures afterward instead of incorporating it in the original design could constitute a problem. For example, the in-vehicle network has historically been a closed environment responsible for the control and maneuverability and safety of vehicles. The in-vehicle network has been designed to provide this functionality and security has not been part of the design. In the connected car of the future, external communication is allowed to interact with the previously isolated in-vehicle network. Thus, the in-vehicle network is opened up to potential attacks. Designing security solutions for the existing in-vehicle network creates difficulties as real-time constraints, protocol and hardware limitations need to be considered. In addition, security solutions must not interfere with the functionality provided, e.g., by imposing delays as this could have serious consequences from a safety perspective. Due to economical constraints it may not be possible to redesign the entire system with security in mind. Either the best possible security solutions considering the existing system are developed and applied and as a result possibly degrading the system's performance, or good enough solutions are applied to ensure that the existing system's functionality is left unaffected.

Usually, legacy systems operate for long periods (10 - 15 years) without needing to update their proprietary networks, since their performance and reliability meet the requirements. This was not an issue while the systems were isolated but now, with the need to exchange information and provide connectivity, the implemented proprietary solutions open security flaws which expose the systems to attacks. The misunderstanding that using old and proprietary solutions protects legacy systems from attacks, since the attackers do not know them, gives false sense of security. On the one hand, determined attackers could study the command structure of the system and then use it when they access the network. On the other hand, with so many specialists being laid off, there is a potential new type of attackers with inside knowledge about these systems [119].

Many legacy systems use Microsoft-based architecture and there are still old versions of Windows in use in these systems which are no longer supported and thus unprotected. Windows 2000 is on extended support through June 30, 2010. Any earlier version will have no more security updates.

Possible solution(s). The short-term solution could be a better understanding of how to best adapt security to such systems. Analysis of what new features can be added without unnecessary risk is needed. Experts recommend [119] to study all connection points in the network, understand what traffic has to flow from the old networks into the business network. Having a very tightly configured firewall is important and the information should be flowed through a more modern server, which can be better protected. In general, analyzing the current architecture in detail and cataloging all software running on the control networks help discovering the weaknesses of the network and strengthening its security.

It is also advisable to study the dependability of the different parts.

New architectures can be developed where security permeates all parts of the design for the long term. Migrating to new technologies, however, takes time, while security is needed at the present moment and this reality could influence the process of introducing new and more secure technologies.

10.3 Use of COTS components

Threats. The use of COTS components and systems can make any system, but especially a system connected to a critical infrastructure, vulnerable to a variety of attacks. There are two problems with COTS components. The first problem is related to hidden functionality and outsourcing, as described in Section 6. The designer has no real control over the product he is introducing into his system. The COTS product is designed (and manufactured) elsewhere and the documentation can be incomplete or even faulty. There is no guarantee that there is no hidden functionality, such as back doors or Trojan horses. Nor can the absence of these be verified, as discussed in Section 6.

The second problem is related to the generality of the COTS systems versus the sometimes very specific requirements of the environments where they are used. It is this second problem we describe below.

To reduce cost and time for design, the use of COTS systems and components in critical applications seems attractive and will thus continue. COTS systems are used in industrial automation process-control systems because they are cheaper and more efficient. Going to COTS components the emphasis is on cost and new operations. In process control systems, however, the main concerns are availability and safety. There is a gap between the priorities (safety versus cheap COTS components) and this gap leads to new challenges to security and reliability.

There are some projects (e.g., DEAR-COTS [29]) where COTS components are applied to design distributed computer-controlled systems. They are organized using redundancy and design diversity to make the system dependable and secure. Some of the issues addressed in DEAR-COTS are the use of emerging information technologies to cope with heterogeneity issues while providing a dependable user-friendly man-machine interface.

Another trend that seems inevitable is the transition to ICT in process control. Proprietary solutions are replaced by open and conventional protocols and networks and security techniques and technologies have to be introduced. There are efforts to apply COTS components and open-source standards along with the standards for process control systems. The organizations from industry that develop commercial interface standards work with some military programs to include real-time and fault-tolerance requirements [154].

A real-life process control system for oil and gas is shown in Figure 10.1 (Courtesy of The Norwegian Oil Industry Association). The objective in this system is to introduce more automation and many ICTs will be implemented into it in the next five years. These systems will be operated remotely. The operator will be out of view of the real systems and it will be difficult to assess any special situations that may arise. For that reason, computers will control many functions but they are prone to virus infections and attacks. There will be remote access through connections to the Internet, leading to new threats. Response management is needed, coping with incidents – recovery, isolation, and restoring the system to a working state. Forensics should also be applied to determine the responsibilities.

Possible solution(s). No good solution exists, but various work-arounds, such as using COTS systems with some fault-tolerant approaches (replication, diversity approach); applying the COTS components in non-critical areas only; introduce and manage heterogeneity; or use of a compact and trusted application base.

Another possible approach is to introduce semantic technologies, i.e., to take a holistic approach to security with semantic technology (e.g., SOA). Physical components should be classified, as they have to be defined from the basis. We have to identify and decide what and how to protect, i.e., an assessment of the assets to be protected has to be done.

10.4 Next generation networks

Threats. Recently, there is a general trend for carrying multimedia in the field of electronic communications. This was imposed by the Internet as it is its inherent feature. Under the pressure of the Internet, on the one hand, and because of the increased service requirements of end users, on the other, some telecommunication companies are migrating to the so-called Next Generation Networking (NGN).

NGN is a broad term describing some key architectural modifications in the telecommunication core and access networks that have been deployed in the last five years. The main goal of NGN is that one network transports all information and services (voice, data, and multimedia) by encapsulating them into packets, as is done on the Internet. NGNs are commonly built around the Internet Protocol

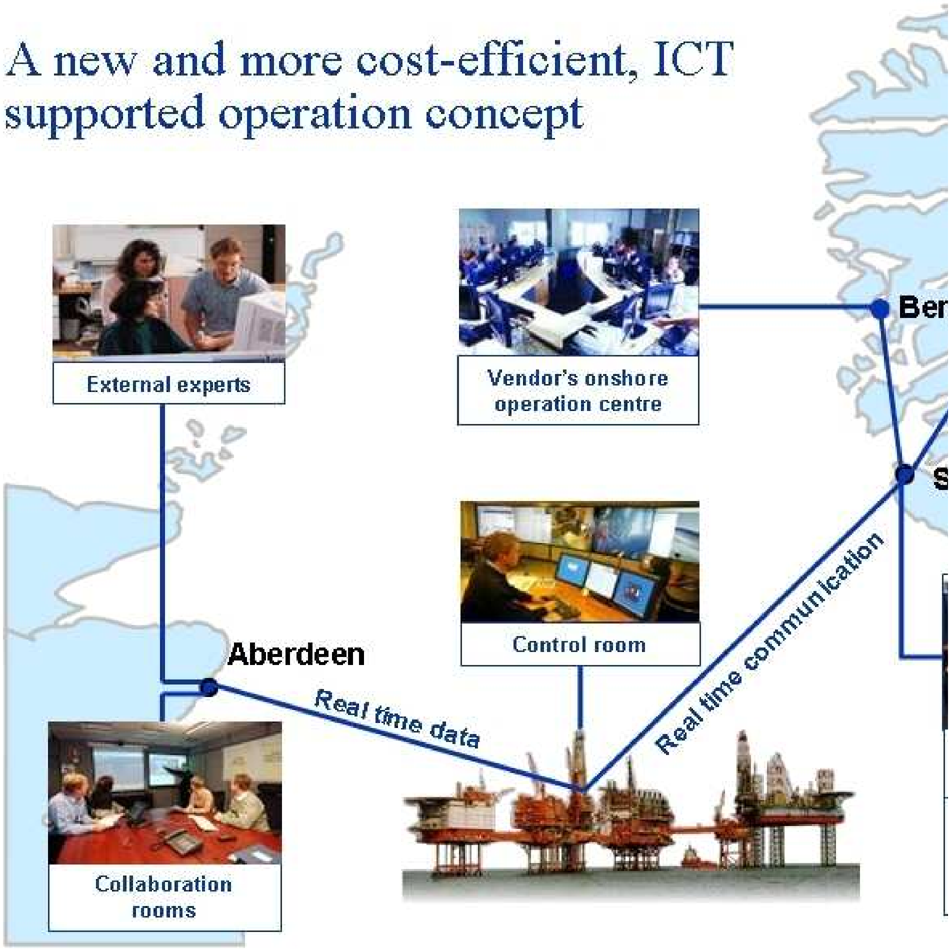

Figure 10.1: A new and more cost-efficient, ICT supported operation concept.

and therefore the term *all-IP* is also sometimes used to describe the transformation towards NGN [155].

Within the Bulgarian Telecommunications Company (BTC), a project has been running since 2004 for migration of the existing national operator's telecommunication networks to NGNs. The BTC has built the so-called *converged NGN infrastructure*, which provides voice services, transport services for VPN, data services, and Internet access services. The general idea behind Next Generation Networking is to combine best characteristics of former and present communication technologies [155, 129]. In addition, similar projects to the one deployed in BTC are going on in KPN in the Netherlands, in Ireland [155] and in British Telecom (BT) *21CN* in the United Kingdom.

The openness and easy access and usage of NGN lead to an increased number of vulnerabilities and extreme attention to security measures must be paid. The following is written in [150].

> As part of its responsibilities, DHS (Department of Homeland Security) created the National Infrastructure Protection Plan to coordinate the protection efforts of critical infrastructures. The plan recognizes Internet as a key resource composed of assets within both the IT and the telecommunications sectors. It notes that Internet is used by all critical infrastructure sectors to varying degrees and that it provides information and communications to meet the needs of businesses, government, and the other sectors.

This excerpt confirms the *upcoming critical role of the Internet in CI* and this trend seems unavoidable. The same basic characteristics, which make the Internet so prone to evolution and so ubiquitous, are now sufficient for considering the Internet in itself a potential threat-generating environment.

Recently, many security experts bring up the attention to the specific vulnerabilities of the NGNs. The most exploited among them are [39, 127]:

- Knowledgeable end users can gain access to the control plane of “all-IP” networks like NGNs
- Large number of external connectivity points (and from any other point/site of the Global Internet)
- Shared core network among several NGN operators (the possibility of occurrence and the variety of vulnerabilities is higher)
- Malicious users can manipulate the traffic more easily as no physical access is required.

More than of 32 fundamental vulnerabilities in NGNs are described as a result of the systematic assessment of NGN vulnerabilities [111].

On an open network such as the NGN, capabilities and responsibilities for providing security may reside at any level/layer or with any participant, making end-to-end security a real challenge. The NGN as part of the information infrastructure, and thus as a critical asset, depends upon transport networks being highly available, reliable and tamper-free, even under stress [146].

Possible solution(s). Security mechanisms on open packet networks will be very different from those of legacy telecommunication services in many aspects. In legacy networks, being circuit-oriented vertical networks, much policy management was “built into” the integrated service, comprising all aspects of the network. Security will need to be addressed differently in the NGN.

The design and implementation of NGN need to meet complex requirements, which complicates its security architecture. As a consequence, it is difficult to use a single standard to define it [163]. As a present security solution it was recommended in [160] to use *multiprotocol label switching* (MPLS) VPNs to construct an NGN virtual private bearer network, and thus logically separate NGN services from traditional data services.

Chapter 11

Introduction: Scenarios

The following chapters contain ten hypothetical scenarios that show how attackers can leverage some of the emerging and future threats to reach their malicious goals.

Each scenario starts with a short *overview* that sets the stage. Then, we introduce a narrative that describes a concrete threat *scenario*. This narrative is in prose, and discusses the motivations and actions of the actors that carry out attacks. It also shows how certain weakness and vulnerabilities can be combined to reach an overarching goal. We have decided to describe concrete scenarios in this fashion as a complement to the more generic listing of threats in the first part, and we hope that this form might provide concrete evidence and motivation to develop adequate security defenses. We also briefly discuss the *consequences* of such attacks on the people, the economy, and the environment.

In addition to the scenario text, we include with each scenario a discussion section (called *could it happen?*) that captures the likelihood of the scenario to turn into reality. Clearly, there are certain scenarios that are more far-fetched, while others are quite near-time and rely on threats that already exist, in one form or another. This section provides evidence from previous incidents or papers in the literature that underline that the scenario is not purely fictional. Also, in some cases, it captures the responses from domain experts who have read the scenarios and provided feedback. The last point is important: We did not invent scenarios in complete isolation, but we consulted with domain experts who were shown earlier drafts of the texts. The feedback and information that we received was then used to improve the scenarios and/or ensure that they are plausible.

Finally, we also relate each scenario to the threats (in a section called *related threats*) that were identified and previously described in the first part of this document. The reference numbers match the entries in the list of threats given in Table 2.1.

Chapter 12

Scenario: Election Fraud

12.1 Overview

Elections are formal decision-making processes by which a group of people (or the citizens of a nation) choose an individual or a party to represent their interests and to hold public office. In the last few years, e-voting has been considered as an alternative means to organize elections. With e-voting, votes are no longer cast by marking or filling out forms with the voters' choices, but by entering the selections into a computer system. Proponents of the e-voting process argue that it is more accurate, faster to tally the votes, more convenient, and less costly. Some argue that it also provides more accountability. However, whenever a process leverages information technology, there is always the risk that ICT threats are suddenly becoming relevant for this process as well. In the case of e-voting, ICT threats can lead to election fraud, where attackers deliberately tamper with the electoral process to modify the outcome of the election. The following describes a scenario in which ICT-related threats manifest in the theft of an election.

12.2 Scenario

The situation is not looking too promising for Dr. Smith and his political movement. The polls have shown a steady decline over the last few weeks, and the most recent numbers based on phone interviews predict a result between 19% and 20%. This is in stark contrast to the almost 30% that seemed possible only a few months ago. Unfortunately, his political competitors have unleashed some successful ad campaigns that have exploited the fact that a top party leader and close friend of Dr. Smith was found guilty of embezzlement and had to step down. Clearly a significant blow to his campaign that was built upon trust and core values such as integrity and honesty. So, Dr. Smith was in a weak situation when the phone rang two days ago, and the caller made an interesting proposal. For just ten million Euros, the person on the phone promised, he would "fix" the problem and lead Smith's party to victory. At first, Smith was dumbfounded. How could that be possible? Clearly, only a miracle could secure an election victory now, and Smith was in politics long enough to not believe in such miracles. Of course, this could just be a joke.

Smith just wanted to hang up, thinking about a cruel joke that one of his political enemies would play on him. However, the situation was desperate, so he decided to at least give that guy a couple of minutes. Interestingly, the caller quickly came to the point. He mentioned that, for the first time, the upcoming election would make nationwide use of e-voting. He further mentioned that he had detailed knowledge of the software that would be used by the chosen e-voting machine vendor, and had a plan how he could tamper with the results on a scale that would allow the results to be turned upside down. Of course, the fraudulent modification would not make Dr. Smith's party win by a landslide, but just enough so that the results remain surprising, but in the realm of the possible. After all, pollsters don't have the best reputation and track records these days.

Smith remembered some debate about whether e-voting would be a good idea or not. Skeptics have always pointed out the possibility of voter fraud, but e-voting proponents and vendors were quick to dismiss such concerns. The software was verified by "experts," and security concerns only scaremongering by people who are against technological progress, they said. Also, e-voting has the promise of being more efficient and thus, less costly. In times of financial problems and budget cuts, how could the government decide against such a proposal. Company lobbyists did their part, and hence, e-voting was introduced with much fanfare.

Smith was still sceptical, wondering why the caller would make this offer to him and not others. The man on the phone explained that he was always sympathetic of Smith's ideas, and that he would love to see the current government replaced. Also, he provided some more details of the plan, and obviously, he knew what he was talking about. Sometimes, Smith still has troubles believing what happened next; when he actually agreed to the proposed plan. Payment arrangements were made, and the caller assured Smith that he needs no longer to worry. He would just need to go along and wait until the big day. He, his party, the pollsters, and the rest of the country would be in for a big surprise.

John always liked to tinker with computers. But he was also lazy, and found that some small-scale computer fraud, such as stealing credit card numbers, was easier than actually working in an office eight hours a day week after week. So, he had already some experience with writing malicious code and infecting people's computers. When he heard about the planned move to e-voting, he was first only remotely interested. However, this changed quickly when he found some leaked source code of the e-voting software online. Apparently, someone at the e-voting company has just forgotten some copy of the voting software on an open FTP server, someone else noticed this, downloaded the program, and put it on the Internet. John studied the code, and was surprised to find that the overall software

If this seems far-fetched, observe that this is exactly the way in which e-voting software packages of both Diebold and Sequoia were leaked into the public domain [161].

quality was relatively poor; definitely not something that one would expect from programs that are entrusted such an important process like an election. In fact, if he could get access to one of these new e-voting machines, he could easily insert a malicious memory card that can take complete control of this machine. This would allow John to freely tamper with the data on this machine, adjusting the votes to his liking. That sounded like an interesting plan.

However, tampering with a single machine can hardly influence the outcome of an entire election, and when a single voting machine would contribute millions of votes to the end result, this would clearly raise some eyebrows. So, this wouldn't work. John was almost ready to give up when he realized that voting machines are actually not stand-alone devices, but they are all connected to the central tabulation server at the Ministry of Interior. This was one of the arguments for e-voting after all; the results would basically be available at the instant the voting booths close, and there is no tedious waiting for hours while votes are counted and no more imprecision due to miscounting by hand or lost bags of votes. Thus, John could use the network to spread his malicious code from voting machine to voting machine, something that computer worms have already demonstrated to be feasible on the scale of the Internet. And in this scenario, all machines are exactly the same, running the same vulnerable software on the same hardware platform. It was an ideal breeding ground for distributing his software.

He still had to get access to a single voting machine to insert his "upgraded" memory card and launch his malware program, but this was easy. Definitely much easier than physically tampering with all, or a large number of, voting machines. John checked out the school next to his home. Rolling out voting machines is not a trivial tasks from a logistics point of view, and thus, the government started to move the machines to the voting places, such as schools and public offices, several days before the election day. Of course, the door to the school building was looked, but this is by no means a building that is built to withstand a determined intruder. In fact, it is not uncommon that kids enter their school after hours or during the weekend, just to prove their courage to others. Thus, John simply waited until the night and checked out the premise. It was easier than expected, the door was not even locked, and so he could simply walk up to the schools cafeteria, where the machine was sitting. John knew the design of the voting machines well enough by now to know what button to press to eject the memory card that holds the voting software. This would be easy to replace with his modified version later on. With a smile, John silently disappeared into the darkness.

Now, he would just need a way to make money off this "project." Of course, he could simply cause havoc and change the election to something that is obviously wrong. But this would just cause a re-run of the election and besides the pleasure of having caused some trouble, there is really nothing in there for him. So, John came up with the idea of offering his services to Dr. Smith's party. He always felt

Again, it might seem unlikely that e-voting machines would be left unprotected, but this is actually what happened in previous elections in the US [36].

sympathetic for Dr. Smith, as he is finally someone who is honest. This resonated well with John, who always felt some latent feeling of unhappiness with the current ruling class. He figured he might kill two birds with one stone; electing the leader that he wanted, and making some money out of this. John bought a pre-paid cell phone and called Dr. Smith. The plan was not really worked out completely, but to John's surprise, it was relatively easy to reach Dr. Smith. The surprise was even bigger when Dr. Smith accepted his proposal. Now, John could carry out the plan. The next few days, he spent coding the actual worm that would carry out the attack at individual voting machines, and he quickly added some extra routines to have the code spread when all voting machines are connected to the central voting server at election day. Once this was finished, John prepared the malicious memory card. The only thing that was left to do was to insert this into the machine that he previously visited at the local school. At this point, all was set, and he just had to wait for the big day.

Finally, election day arrived, and both John and Dr. Smith are waiting anxiously until the voting booths close. Then, at 6pm, both stared at the TV screen when the news reported switched to the central election office. No need to wait for hours until the votes are counted, and no need to rely on exit polls and projections. With the press of a button, all votes would be tabulated, and the final results displayed on the screen for everyone to see. The shock and surprise that set in at this point was almost too much for the reporter who desperately attempted to keep her cool. This was a huge surprise, and nothing predicted by previous polls. Dr. Smith party came in first, with a small margin of less than two percent, but still the clear and unexpected winner of today. Dr. Smith smiled as the camera turned towards him. Clearly, this guy was worth his money!

12.3 Consequences

The consequences of election fraud are significant and undermine the trust of the population in the elected leaders and their legitimation. Moreover, when people gain access to legislative and executive power, they have significant direct and indirect influence over the society and economy of an entire nation. Of course, they can also influence other nations through trade, negotiations, or military force.

12.4 Related threats

The general threats related to this attack scenario are related to *hidden functionality* (Threat #15), *new vectors to reach victims* (Threat #16), *advanced malware* (Threat #18), and, possibly, the *insider threat* (Threat #13). Clearly, the election fraud works because attackers implement hidden functionality into e-voting machines. Such attacks would be launched with advanced malware, possibly taking advantage of insiders that could be part of the electoral process or the e-voting company. Finally, attacking e-voting machines also means that attackers use new means and vectors to reach victims and achieve their goals. When a criminal organization is involved and uses the fraud scheme to make money, the attack could also be considered to be fueled by the *underground economy* (Threat #3).

12.5 Could it happen?

The adversary in the scenario outlined above was a single person, and hence, relatively weak. It is conceivable that political parties and/or criminal organizations would have significantly more resources to carry out an attack as outlined above. Moreover, as mentioned in the footnotes to the scenario, several reports have confirmed that e-voting software is vulnerable, and attackers could trivially gain access to voting machines.

Two aspects make the scenario outlined above more difficult in current real-world settings. First, e-voting machines are typically not connected to a central tabulation server. Instead, after a voting place closes, the memory card that is inside a machine is taken out and transported to a central location, where the votes stored on each card are read out and tallied. Thus, it is not possible to directly spread via the network to other voting machines. However, as shown by previous studies of e-voting system deployments in California and Ohio, it is still possible to first infect a single memory card, and later use this infected card to spread to the central node where each memory card is brought to.

A second problem are possible countermeasures deployed by the e-voting machine, for example, a paper trail (or print-out) that provides evidence of the actual votes that were cast. Again, the malicious software author has a number of ways to make these countermeasures less effective (or even ineffective) so that large-scale voting fraud is possible, as described, for example, in [14].

Interestingly, as result of actual e-voting system evaluations in the US, several vendors were actually de-certified or had to improve their platforms. This shows that threats against e-voting systems are real, and serious security problems were found. Although, up to now, no significant e-voting-based manipulations are publicly known, the threat is clearly real and requires additional effort from the research community to protect the integrity of the electoral process.

The scenario was checked with experts who have participated in e-voting security audits in California and Ohio. The experts have confirmed that the scenario outlined previously is quite unlikely but possible. In particular, it is unlikely that a single person can commit large-scale election fraud. However, a more powerful adversary (such as a criminal organization or a political party in a close race) with more resources could plausibly stage such attacks and manipulate the outcome of elections.

Chapter 13

Scenario: Crashing the Stock Market for Fun and Profit

13.1 Overview

It has been reported that a third of all EU and US stock trades in 2006 were driven by automatic programs [6]. As of 2009, high frequency trading firms account for 73% of all US equity trading volume [3]. The trend of using automated trading systems is to continue and bond markets in many countries are moving toward more access for algorithmic traders.

Clearly, the more our financial infrastructures become dependent on computerized systems, the more vulnerable our financial system becomes to cyber attacks. In this scenario, we describe how organized cyber-criminals leverage the fact that many stock markets are dependent on automated trading (i.e., algorithmic trading) and buying systems in order to crash the stock market and make a financial profit.

13.2 Scenario

Vladislav Doronen is a 25 year-old computer scientist who has studied at the Molayew Technical University. He is a motivated, very talented young man who has moved to Vienna, Austria in 2015 to work for an IT company there. With the increasing role of IT systems in all aspects of European life, the demand for IT experts has grown, and EU governments have been more than willing to open their employment markets to qualified IT experts from all over the world. The company that Vladislav is working for has expertise in creating, maintaining and selling automated trading systems for the Austrian stock market.

On March 24^{th} 2017, while attending a conference at the Technical University of Vienna, Vladislav is approached by an individual who introduces himself as "Dieter." Dieter tells Vladislav that he is very interested in automated trading systems, and that he is in the process of creating his own company. He tells Vladislav that there might be some very interesting employment opportunities for him and that they should meet up for lunch.

Vladislav is interested. After all, he is young, he is very good in what he does, and a higher salary is never a bad thing. In fact, he has had the feeling that his current company has been exploiting him and that he has not been receiving what he deserves.

On March 28^{th} 2017, at 12.30pm, Vladislav and Dieter meet for lunch and go to a small Italian restaurant near the Naschmarkt. Dieter tells Vladislav that Vladislav's expertise would be invaluable for a project that he and his "friends" have come up with. If he agrees, Vladislav can become a very rich man, Dieter tells him. Vladislav share in this project would be one million Euros . . . however, naturally, he is not to talk to anyone about this. If he does talk, Dieter tells him, his powerful "friends" will not be happy, and that they tend to become physically aggressive sometimes.

Vladislav is not too happy for having become involved with people who, apparently, are threatening him physically. However, the thought of having a million Euros is too tempting. He looks at Dieter and says: "Ok, I'll do it. But I want 250 thousand Euros in advance." Dieter agrees.

Dieter and his friends have come up with a sophisticated plan of how they can make a lot of money in a short amount of time. The idea is simple: Crash some stocks artificially, and buy while everyone is selling. The key insight is that the stock market will recover soon, and this will allow Dieter and his friends to sell their acquired stocks with a high profit. The implementation of the attack, however, requires some delicate planning and in-depth technical knowledge of trading systems.

Vladislav has been programming stock market trading systems. Hence, he is knowledgeable in this area. Furthermore, he is a vital asset for Dieter and his friends as the system that Vladislav has been involved in has been sold to many companies that trade stocks at the Vienna stock exchange.

Dieter, Vladislav, and their friends make a four-phase plan. In the first phase, Vladislav is to write malware that can specifically attack and manipulate the automated trading system. A simpler approach would have been for Vladislav to create a backdoor in the automated trading system. This strategy would have been too obvious, though, and would have made it easy for the officials to trace the attack back to Vladislav.

In the second phase, Vladislav is to distribute the malware to the companies that are using the automated trading system.

In the third phase, the malware is to kick-in on a specific date that Dieter and his friends have chosen and start manipulating the automated trading systems. On that date, all automated systems are to automatically start to sell the specific stocks that Dieter and his friends (who are, apparently, knowledgeable in stocks) have identified.

In the fourth phase of the attack, the attackers are to start to buy stocks whose values have considerably decreased. The idea is that when all the automated trading systems start selling the stocks at the same time, the stock market will be affected and some stocks will crash. Automated trading systems use different algorithms and one of these algorithms looks at the trends in trading. For example, if there are sharp decreases in some stocks, the trading system might decide to sell quickly in order to limit the financial loss. By leveraging such algorithmic properties that Vladislav is aware of, Dieter and his friends expect the plan to work.

Vladislav helps prepare USB sticks that contains the malware. These USB sticks are sent to individuals in the companies that Vladislav has identified. By inserting the USB stick, the malware is automatically activated and installs itself on the victim machine. It then starts to look for credentials and components that would allow access to the automated trading system.

150 USB sticks are sent to individuals who work in stock trading companies that Vladislav has identified. A special letter is prepared that references each individual and contains details that only this person can know (e.g., here are the pictures of the X-mas party last year). When inserted, the USB stick installs the malware, deletes itself, and prints out an error message so that no suspicion is raised.

On June 17^{th}, the remote servers that Vladislav has set up using a bullet proof hosting service is receiving responses from 82 installed malware instances. The USB attack has succeeded for some of the victims. Vladislav is able to look at the contents of the victims' machines, and access some of the automated trading systems that these individuals are connected to.

On July 22^{nd} 2017, a bleak economic outlook report is released by a Wall street institution. The Austrian stock market, as well as many stock markets around the world are affected. The stock market shows a tendency to go down. Dieter and his friends decide to use this opportunity to go ahead and launch the attack. Vladislav is instructed to manipulate the automated trading systems so that they start to massively sell specific stocks. By starting a selling frenzy, Dieter and his colleagues manage to create an environment of uncertainty. A cascading effect is the result of the selling frenzy the automated systems have started, and the stock market starts to strongly lose value. Dieter waits for the stocks to reach a low value. When this value is reached, he tells his friends that they should start buying. Vladislav is instructed to stop the automated selling frenzy.

On July 25^{th} 2017, the stock market starts to rebound. Dieter and his friends have managed to make a nice profit with their sophisticated attack.

13.3 Consequences

The immediate consequences of the above scenario include a significant financial loss, damage to reputation of affected companies, and a loss of trust in an important European financial institution.

13.4 Related threats

The scenario that we have discussed is closely related to a number of threats in the malware and fraud domain, such as the *underground economy* (Threat #3), *advanced malware* (Threat #18), and *targeted attacks* (Threat #26). Also, the attack leverages threats due to *cascading effects* (Threat #9) of tightly coupled (trading) systems. In the scenario, the attack also relied on the fact that Vladislav was an *insider* (Threat #13). To summarize, malware is a suitable weapon to perform financial fraud today. With the appropriate insider knowledge about a domain, an attack such as the one we described in this scenario becomes feasible.

13.5 Could it happen?

While the attack may sound like science fiction, unfortunately, the threat that automated trading systems may be manipulated in the future is feasible.

In the stock crash of 2008, we have seen the negative role that automated trading systems have played. As these systems are programmed to reduce loss, a cascading effect caused many automated systems to start panic sells. As a result, normal investors in the market were also negatively affected and started to sell as well.

As IT systems become more and more integrated in our daily lives, there will also be an increase in the numbers of malicious individuals who understand and have the means to manipulate these systems.

Clearly, the systems that we deploy for critical operations need to be trusted. A trusted system could have prevented the attack that we described in this scenario by blocking the installation of the malware.

Unfortunately, trusted computing is still in its infancy. In the near future, we expect virtualization technologies to evolve and become more sophisticated. Using such technologies, it should be possible to create platforms that are trusted and that can block attacks launched by malicious applications.

Chapter 14

Scenario: Industrial Espionage

14.1 Overview

This scenario describes how the foreign intelligence service of an adversarial country uses custom-tailored malware to spy on and eavesdrop on the e-mails and instant chat messages of the development engineers of a fictional European company. This European company is located in France and is an important technology provider for several European armies. The company, DALES, specializes in the development and production of remote-controlled aerial drones. These drones are able to monitor a designated area for up to 24 hours and provide satellite-based real-time video feeds, sound recordings, and infrared pictures. Furthermore, if needed, the drones can also be used to launch laser-guided rockets against stationary or moving targets. Clearly, the products developed by DALES are state of the art, require a significant development effort, and are of high interest for many countries around the world. The drone technology is expensive, but provides a strategic advantage in many modern operational fields (e.g., the fight against insurgency, terrorism, drug trafficking, border surveillance).

We will call the fictional, adversarial country in our scenario PRK (People's Republic of Keko). Because of the strict European export regulations for key technology, PRK is not able to buy the drone technology from DALES. Furthermore, PRK is mainly interested in acquiring, understanding, and reproducing the technology by itself rather than buying it from another country.

14.2 Scenario

On February 12^{th}, 2011, Mr. C., who is the head of the PRK foreign intelligence service, CKKK (Chinowska Kapara Kopa Keko), is summoned by the president. The president informs Mr. C. that he is not happy about the way the negotiations are going with DALES in acquiring the drone technology. DALES is willing to sell a downgraded version of the drone that has a limited aerial surveillance capability. That is, the drone is able to monitor the designated area for only six hours, and is to be fitted with second generation electronics rather than the more advanced third generation electronics that DALES possesses. Furthermore, the drones are to be equipped with standard engines, and not the efficient silent engine technology that DALES has developed. The silent engine technology is important for the PRK as drones that are fitted with standard engines can be easily spotted and shot down.

The president of PRK informs Mr. C. that the PRK needs the advanced drone technology in the fight against the insurgency in the south of the country. Mr C. is to do everything in his power so that the PRK can acquire the advanced drone technology. Mr. C. and the president decide to call this operation, "Operation Yara." Yara was a folk hero in the early 20^{th} century while the PRK was a European colony. Yara started an insurgency against the European powers, and was able to equip his army with self-built rifles.

The following day, on February 13^{th}, 2011, Mr C. summons his cyber-infiltration team for a meeting to his office. This team, known as "Team Dos" in the CKKK, is specialized in vulnerability exploitation and sophisticated Trojan horse development. Team Dos is assigned the task of infiltrating the IT infrastructure of DALES and obtaining sensitive information about the drone that would allow the PRK to rebuild the technology.

Operation Yara is to be conducted in three main phases:

1. Information collection and vulnerability identification.
2. DALES infrastructure infiltration.
3. Appropriate Trojan development and deployment.

Team Dos initiates the phase one of the operation on February 15^{th}. After a one day preparation, a slow vulnerability scan is started that targets all public services of DALES. The mail servers and web servers of DALES are contacted and the scanners look for known vulnerabilities in the products that DALES uses. The scan is performed slowly in order not to raise any suspicion by the administrators of DALES.

On February 20^{th}, the slow scan of DALES is completed. Unfortunately, but not surprisingly for Team Dos, no known vulnerabilities are discovered in the DALES online public services. Because of the sensitive technology that is involved, apparently, DALES has been taking special measures to make sure that its public services are free of known vulnerabilities.

Team Dos resorts to manual scans and decides to look for vulnerabilities in the public services that are not widely known, but that may have been introduced by a developer who was not careful.

On February 22^{nd}, 2011, one of the members of Team Dos discovers a second-order SQL injection vulnerability in the phone directory application of DALES. Apparently, this application is custom-tailored for DALES and has been written by the IT department of the company. The SQL injection vulnerability allows Team

Dos to open a reverse shell to a compromised home machine in France, which Team Dos will be using as a stepping stone. They open the shell by using the standard, default stored procedures of MS SQL Server that allow commands to be invoked in the underlying operating system (Windows NT in this case).

The reverse shell gives Team Dos the opportunity to poke around the network and to look for new venues of attack. They discover that the web and the mail servers have been placed in a demilitarized zone. This is good practice as it protects the internal services of the company if the public services are compromised (as in this case). As Team Dos is interested in intercepting the communication within DALES, the next target is determined to be the mail servers that are located in the same subnet as the web servers.

One of the members of Team Dos discovers that he can access the mail server as root as the web server is able to access the mail server via an SSH key. The IT people at DALES have probably enabled this access as port forwarding is necessary from the web server to the mail server to allow the employees to read their mails via the Horde Webmail application.

Once on the mail server, Team Dos discovers that they can read the e-mails of some of the employees, but not all. DALES is a large company that has thousands of employees. Hence, the employee e-mails are not all stored on one server, but are distributed across several subnets and locations.

Team Dos starts intercepting the e-mail communication (i.e., by regularly uploading the INBOXes of the employees to remote, compromised servers that serve as stepping stones).

On March the 2^{nd}, 2011, Team Dos discovers that the e-mails of two systems engineers of the drone project are stored on the mail server that they have compromised. They discover that the two systems engineers have received invitations for a meeting with their project manager on March 4^{th}.

Team Dos finally has the opportunity to launch a social engineering attack. A fake e-mail is created that is supposedly coming from the project manager. The spoofed e-mail contains a link to a website that contains a drive-by download. Team Dos knows that DALES is using Internet Explorer (IE) as the company browser. Furthermore, they have access to a zero day exploit for IE, which another team had developed several months earlier.

The spoofed e-mail references the planned meeting, looks like a “reply” message from the project leader, and says that everyone should check the link and the website before the meeting. The website looks harmless and contains a calendar with the date of the meeting. However, in the background, a drive-by download attack is automatically launched using heap spraying.

In the morning of March 3^{rd}, 2011, Team Dos has been able to successfully exploit and compromise the desktops of the systems engineers. Because the social engineering attack referenced only facts that the two engineers could know, none of the engineers became suspicious, and naturally, clicked on the link.

As of now, Team Dos has access to the desktop machines of the system engineers. They are able to start sniffers on the subnet where the engineers are connected. The information is leaked out over port 80 with a Trojan horse that piggy-backs onto Internet Explorer web requests. As the employees are allowed to browse the web, the firewall does not filter out these requests. Furthermore, by using a pull-based, web command and control infrastructure, Team Dos is able to send remote commands to the Trojan.

As a positive surprise, Team Dos discovers that the system engineers are able to access many local drives where sensitive information about the drone is stored. They can now simply access these drives, and upload the information to remote servers of their choice. As the Trojan has been specifically written for the attack against DALES and has been obfuscated using modern packers, the anti-virus scanners on the engineers' desktops do not discover it.

On April 2^{nd}, 2011, during a standard security audit, the internal security department of DALES discovers the Trojan horses by chance. The Trojan has lost connection to a remote server, and because of a bug, has started to loudly scan the internal network.

Although this incident and the early discovery of the attack is unfortunate for CKKK, the organization was able to collect enough sensitive information about the drone during the month that it had access to the DALES systems. Many design documents, e-mails, instant chat messages, and blueprints were stolen. This information gives a major push to the secret drone development program of the PRK.

14.3 Consequences

The immediate consequences of the above scenario include financial loss, damage to reputation, threat to European security, and a significant negative impact on European interests.

Note that the threat is more general than the specific scenario that we sketched above. Such an attack can be launched against organizations that do not only deal with technology. The scenario can also be used for stealing sensitive information for conducting negotiations, damaging individuals and competitors, and injecting false information to cause confusion, panic, and havoc.

14.4 Related threats

The threat discussed in this section is closely related to the scenario that describes an attack to the banking infrastructure. Similar to the previous scenario, related threats are the *underground economy* and its support structures (Threat #3), *advanced malware* (Threat #18), and *targeted attacks* (Threat #26).

Malicious code is a significant threat that can manifest itself in different forms. It can be used as a weapon and enabler in many different scenarios, and has the potential to harm a high number of organizations, and individuals. However, malware does not always have to be the main cyber-attack. Rather, it can be the final, powerful step in a general purpose attack as discussed in this scenario. When the adversary is able to install malware on the sensitive infrastructure of the victim, she has gained a major advantage.

14.5 Could it happen?

The scenario that we have sketched is very realistic. We have used real attack vectors, real vulnerabilities, and exploitation techniques that are currently being used in practice today.

Because of the sensitive nature of espionage attempts, detected attacks are often not leaked to the press. However, there have been similar incidences in that past that have received media coverage [25, 30]. The threat, hence, is realistic and appropriate responses are required.

As an interesting update, *after* the writing of this scenario, a very similar, real-world attack was made public in mid-January 2010 [78]. In this attack, referred to as "Operation Aurora," China's intelligence has allegedly infiltrated a number of western companies, including Google, and stolen sensitive documents. Incidentally, the attack also used a zero-day exploit against Internet Explorer.

One of the main problems today is that many Windows-based machines still run with administrator privileges. As a result, it becomes easier to install malware on these machines that can then have unlimited access to the entire machine and the network. Although newer operating systems such as Windows Vista and Windows 7 do not run as system administrator by default, system privileges are still often required in development environments. Furthermore, many companies have been reluctant to switch to these newer operating systems because of considerations such as the increased cost, the necessity to maintain legacy software, and the required training (i.e., the newer systems have different security settings and a similar, yet different user interface).

Our increasing dependence on the Internet and IT infrastructures means that cyber systems are becoming increasingly complex, and difficult to maintain. As the scenario demonstrates, a small vulnerability in an uncritical application (e.g., a phone directory web application) can potentially be used as a stepping stone to launch a more serious, severe attack.

Attacks today are typically composed of several phases. Once an adversary gains access to a vulnerable component of a system, the attack often does not stop there. The attacker can acquire new knowledge about the compromised environment, and use this knowledge to discover and exploit new vulnerabilities.

Targeted social engineering (i.e., phishing) is a threat that has increased in importance in the last couple of years. There is good reason to believe that the numbers of such attacks will continue to increase in the near future. If detailed information can be used in a social engineering attack that only the receiver is supposed to know, a social-engineering based attack has great potential to be successful even if the victims are technically sophisticated. Our scenario shows how such, non-public information can be acquired in practice, and how it can then be used to launch a more sophisticated, targeted attack.

Clearly, the less vulnerable the systems of an organization are, the less probable successful attacks are going to be. Note that the fact that the systems of an organization are secure does not necessarily mean that there will be no attacks. In fact, in reality, there will not be a reduction in the numbers of attempted break-ins.

Techniques and tools are required that can allow us to discover and fix vulnerabilities in IT systems. There is large room for improvement in this area with respect to research. Unfortunately, the available tools are not sufficient to find and fix all simple vulnerabilities such as SQL injection, XSS, and parameter injection.

The problem needs to be tackled in the design phase. Currently, most techniques are trying to find and fix vulnerabilities in the implementation phase. The fact that languages such as Java and C# have eliminated buffer overflow problems by design is a good indication that we can do better in avoiding implementation-level vulnerabilities.

Chapter 15

Scenario: The Smart Grid

15.1 Overview

In this scenario, we illustrate how several of the threats identified by the FORWARD working groups can be used by a criminal organisation to easily take over the smart grid in a city. In this case, the group has not decided on the final use of the takeover, but before they can make use of the infiltrated system, a relatively minor problem has *cascaded* and caused a major blackout.

15.2 Scenario

The night is cold, with ominous signs that the temperature may drop even further below zero. The few people who dare to challenge the biting cold are walking briskly, covering their faces with their scarves as best as they can. Regardless of how warm they dress, there is no escape from breathing the cold air.

Nobody is paying much attention to the surroundings, and even so, it would be easy to miss the white van parked on the side of the street. Nothing from the outside would point to anything out of the ordinary. A single glance on the inside, however, would probably have aroused the curiosity of a few of the pedestrians. Why would a van be parked in this quiet residential neighborhood, with three young men inside looking intensely at a couple of laptops connected to some sort of antenna? They could be servicing some system, but they do not look like typical repair men. Most of the passersby would think nothing more of the van and the men. A combination of the cold and planning for the innumerable errands to be done before Christmas would cause them to forget all about the white van as soon as they had passed it.

But on the inside, the tension made the van seem smaller than it really was with the sweat-tinged air seeming to contain far too little oxygen. Tonight's operation was important, and they had already been working hard for quite some time. The success would lead to both personal recognition and financial reward. The three men, members of the criminal syndicate Snooze, were about to install rogue software on a smart grid meter in the neighborhood. The short-term goal was simple: install the software to gain complete control of the operation of the meter and then spread this software further by giving it the appearance of an official update. Precisely what the longer-term goals would be were still under discussion. Even though the smart meters could be used as a platform for deploying attacks, there were those in the syndicate who argued that more could be gained using the technology covertly. Apart from the simple operation of changing the billing statements, one could possibly also use the ability to control the power grid in the city in conjunction with more traditional criminal activities. Turning off the right switch at the right time would make any burglary simpler and access to such a service could also, for the right price, be sold to extremist groups for them to make use of as they wished.

The silence in the van was suddenly interrupted:

– It won't work, said Adam. They've changed the ZigBee chips to the second generation. We can no longer sniff the key. I told you we should've done this a month ago. This part of the city was last on the upgrade plan. If the attack did not work here, it would not work anywhere else in the city either. Adam looked at the laptop as though it was its fault that their plan would not work.

– We knew this was a risk, answered Caleb, but we had little choice. A month ago, our firmware was far from ready for deployment so it wasn't like we really had any sort of option. Honestly, I'm still not sure it's ready but we agreed that this would be a good compromise between having okay firmware and the possibility of this part of the network still being vulnerable. This really must be the first time Snake Electricity has done anything on time, said Caleb, slowly shaking his head.

– Well, said the last person in the van quietly. We have a Plan B. The question is whether it's worth the risk or if we should postpone.

The three men looked at each other. Plan B involved two of them acting as representatives from the local electricity company and thus gaining physical access to the smart meter. The new firmware could then be installed and the key to the network extracted, but there would always be the risk of discovery or, later on, the risk of recognition. The potential money, however, sang its alluring song and the three men came to a silent agreement. They would go for Plan B.

Thirty minutes later, two of the men left the van and walked up to a house across the street to ring the door bell. According to public records, an elderly woman lived here by herself; the reason why this particular house had been chosen. Hopefully, she would be by herself and not question their authority to access the hardware. When the old woman opened the door, they presented themselves as local servicemen who needed to do a quick test on the power for the house. A central system, they claimed, had registered a small anomaly and they needed to make sure that there was no serious problem. Who would want to pay more than necessary for their electricity, they joked.

As expected, they did not even need to identify themselves with their fake credentials, but were instead invited in and offered a cup of coffee, which they politely declined. Ten minutes later, the men left the house with their mission accomplished. They now had control of one meter in the power grid.

Back in the van, Caleb initiated the sequence to replicate the firmware over the air. He carefully monitored the first duplication but it went without any problems. The new firmware seemed to be functioning as well as the old one did. Satisfied, they pulled away from the curb and drove off. They would now be able to monitor the progress of the self-replication from any part of the city-wide network for the power grid.

As the night progressed, the streets gradually emptied of people. The earlier promise of this being a very cold night was gradually fulfilled. The temperature started to drop one degree after another. It was then that it happened. The low temperature caused the heating system of one particular house in a different part of the city to where the "repairmen" had visited to increase its power consumption. This in turn increased the load on the smart meter in the house making a variable go out of bounds in the rogue, relatively untested firmware. The error caused the meter to communicate illegal values to its neighbors, which in turn caused them to fail. In neighborhood after neighborhood, the smart meters crashed and with them, the city was plunged into darkness: a very cold darkness.

15.3 Consequences

In this scenario, we explained how some individuals connected to a criminal network tried to exploit vulnerabilities in the smart grid. The example is interesting because the power grid is fundamental to many important services in society. It may be used for heating or for cooling in extreme temperatures. It is used to power alarm systems and pumps that give us water etc. The grid is also relatively fragile, in the sense that some minor disturbance can cascade and cause major interruptions [152]. A big push is currently underway to make the grid "smarter" by incorporating information and communication technology. Exactly what this will entail is still up for debate, but the first generation of changes sees power companies interested in facilitating remote access to give them the ability to disconnect customers and read a property's current power consumption. However, domain experts have emphasized that the smart grid is much more ambitious than simple energy meters and may be the future basis for electric energy dispatch. For that reason, the meter should probably be viewed as a "router of energy packets" within the grid. Such functionality will of course affect both the scenario and the outcome. The full implications of such future changes are not covered here.

The consequences of the loss of power have been documented elsewhere [75, 76]. A government agency in Sweden, Krisberedskapsmyndigheten, has analyzed critical dependencies [76]. The supply of electricity is important for a number of important sectors in society for example: cash payments, credit payments, the food sector, sewage, transport, fuel supply, primary care and care of the elderly amongst others. Further dependencies exist, for example within the communications industry although power reserves can, in the short term, minimize such consequences. A long term power blackout will, however, seriously affect services such as cell phone communication. Further disruption for the population (including business and agriculture) will be felt through the loss of electricity for heating, lighting and listening to or reading the news.

15.4 Related threats

The antagonists first tried to use issues in relation to *wireless communications* (Threat #8). In this particular scenario, the first attack failed. The antagonists then take advantage of problems discussed in *sensors and RFID* (Threat #17). The major blackout is caused by cascading effects, an issue discussed in *unforeseen cascading effects* (Threat #9) and partly in *threats due to scale* (Threat #2). We also note that for these kinds of systems a focus on security may not have been so important, something that goes back to the cultural difference between the concepts of control and security. In most organisations safety has always been the priority and security is still not as well understood (Threat #25). Finally, the attackers are also able to use the gullibility of home users and their lack of understanding of security (Threat #26).

15.5 Could it happen?

We consider the described scenario to be a most realistic one. Criminal groups are already exploiting vulnerabilities in the cyber world. Even though they have, thus far, concentrated on regular malware and phishing, it is likely that they will begin considering other types of attacks, especially those directed at critical infrastructures.

The power grid has for a long time been fairly unsophisticated but this is changing. The push for a smarter grid is seen by many as a long-term means to facilitate the connection of greener power-generation technology to the grid. Taking into consideration the much discussed risk of climate change, some people argue that it is necessary to make such upgrades. The more immediate benefits, however, would include a more robust grid and the ability to control the smart meters remotely, for example, to turn off the power in the event of unpaid bills or to measure power consumption more accurately. Many European countries are regulating the use of smart meters, with Sweden and Italy being early adopters. In Sweden, government regulations have made smart meters obligatory. Furthermore, Gothenburg, the second largest city in Sweden, is one of the first cities in the world with a city-wide ZigBee network connecting the smart meters in a network mesh.

In our threat report [40], we have discussed problems in *wireless communications in critical industrial applications*. As a particular instance of these problems, one can look at these ZigBee networks. In the first generation of ZigBee chips the key could easily be extracted. In the second generation, Goodspeed [43] demonstrated how the key could be extracted with physical access to the chip. Hopefully, the third generation will have better security but second generation chips have already been deployed. In the scenario described, the attackers were given physical access to the chip, thus increasing the probability that they would be able to extract a key or update the firmware on the chip.

As a practical example of problems related to *sensors and RFID*, we can consider the smart meter. Like ZigBee chips, the smart meters of today are also not built with security in mind. To focus on a particular exploit, we refer to the work presented by Davis at Blackhat 2009 [28]. From his presentation, it is clear that some of these smart meters have insufficient hardware to ensure adequate protection. In his example, he showed how a smart meter could be hacked and then how, by using self-replicating code, a larger portion of the network could be compromised.

Both the work by Goodspeed, Davis and others show that the current generation of technology has security problems but it is nevertheless being deployed and the necessary upgrades could feasibly cost so much money so that they become postponed for a while. A criminal group with sufficient resources could thus cause problems for a city's smart grid and it is easy to imagine, given its sensitive nature, how a minor problem could *cascade* through the system and cause a major blackout.

One of the major problems is the early adoption and deployment of sensors that are not mature enough from a security perspective. Given that *post-hoc* updating of critical infrastructure is expensive, it may be that we will have to accept vulnerabilities for a long time if these first generation systems are deployed on a wider scale. From a security point of view, a more mature platform is clearly critical in order to avoid the scenario described above. For example, a possible solution may be found by using cryptographic techniques for the access to hardware and software systems. In our threat report, we have discussed the risks with both sensors and wireless traffic. Some security problems are intrinsically linked to the nature of the platform and the medium of communication. For example, being given physical access to a platform often leads to an easier compromise of the system. For that reason, improved physical security mechanisms would present a relevant area of research. Another part of the solution must also involve educating customers about risks. We have also described the problems of complex systems and how errors may cascade through such systems and cause wide-spread problems. In order to better understand such issues, more research has to be put into modelling systems and security, taking into account hierarchical structure and error propagation. For example, compartmentalization of the system would most probably have reduced the damage done. Finally, finding good methods for verification and validation of the actual security achieved is always pertinent and would allow system owners to really know how secure the system is instead of just guessing and finding out flaws in retrospect.

Chapter 16

Scenario: The Oil Spill

16.1 Overview

There is sometimes a culture clash between security versus safety in critical industrial applications. Many times, however, security and safety are just different sides of the same coin. In this scenario, we focus on a large industrial accident. The chosen domain is an offshore drilling platform, but a similar event could also be possible in other, similar areas. The antagonist uses a series of steps to achieve his objective. He takes advantage of a trusted site where he installs malicious software that does not spread actively, but propagates only when the victim host initiates its own connection to this server, thus working much the same as current malware in the web domain. The malicious server software is tailored towards the victim environment and the antagonist also has an insider's view and knows its weak points, i.e., vulnerable routers that can be corrupted, as well as the means to grant himself sufficient authorization for the task (known passwords).

Experts with local domain knowledge stress the complexity of these kinds of organizations. The structure cannot be reduced to the level of considering those inside an organization as "trusted" and those outside as "untrusted." Instead, those working with security solutions must consider a much more finer-grained division between different entities that need to communicate.

16.2 Scenario

This particular October 5^{th} felt like any ordinary day. The sun was shining, the air had a quality to it that could best be described as crisp and the wind was refreshingly strong, mixing the smells of the local beach with the faintest trace of a vast ocean. It was almost ironic, how superficially similar this day was to the very same day five years ago. Similar that is, so long as you forced your gaze towards the horizon, avoiding even a glimpse of the beach at your feet. When letting one's gaze wander across the neighboring surroundings, it was impossible to ignore the visible signs of the oil spill. Sure, it was late in the season, which partly explained the lack of birds. Sure, the desolate cliffs could look normal to a tourist visiting the national park for the first time – for this was, after all, an island in the North Sea and not some fertile paradise in the tropics. For someone who had come here for almost every summer since he was ten, however, the difference was striking. It was as if Nature herself had written across the beach in large letters that this was his fault. Declared the first Swedish national marine park in 2009, neighboring a similar area in Norway, the area was set aside to protect the welfare of 6,000 different marine species. But the walls man built to protect could also be torn down and destroyed. Now, the national park was rather a sign of how impotent legal documents could be against some catastrophes. The thousands of dead birds had found no comfort in those pieces of paper. The fishermen had not been helped by the legal phrases. The whole economy of the region had taken a turn for the worse and understandably so, because what sort of tourist wants to visit a contaminated beach?

He had never thought he would actually be the villain in this story. In his youthful zeal, he had instead imagined himself as the protagonist that would save Mother Nature and teach multinational corporations a lesson. Even though he had managed to stir up quite a debate, it was definitely not worth the final price. Reminiscing, he considered what he counted as the beginning of the story; the day he stepped on the helicopter to fly to the offshore platform. In some ways, he guessed, it had all started much earlier. Growing up and finding a cause to fight for. Listening to the presentation by Al Gore that opened his eyes to the risk of major climate change and his disappointment with the intergovernmental agreements. Kyoto had not helped as major players decided not to participate. Copenhagen 2009 did not turn out any better, and he felt a need to do something truly significant. And then the idea. His idea that had appeared so simple and so potent. By turning off the oil production at an offshore platform, the world would realize the inherent frailty of the current oil-driven economy. The goal had been twofold. Firstly, any disruption would show the weaknesses in the system. Secondly, if the disruption could be made to last a bit longer, it might even hurt the oil company from an economic perspective. Even he, with his inexperience, had realized the second part was improbable – manual local intervention would simply override his remote commands – but there was always hope...

The day he stepped into the helicopter to fly to the platform, he was broke. He had no idea that the simple step into the helicopter and the experiences from the next couple of months would lead him on a road of stupidity that led to this environmental catastrophe. The work had been quite hard physically, but he had found it frightfully isolating. The Internet connection had been his gateway to the world and his anchor to normality. He had always been a whiz kid around computers, so what he learned at the platform, he remembered and later used for his attack. The local router, for example, was the last line of defense against outside traffic, but it still had the default password and could easily be reprogrammed. The segregation between the network for the employees and the critical system was good, but a couple of mistakes had been made. Someone had probably built a bridge for convenience, but it was a path he had been able to use. The personnel at the platform had also been extremely competent in doing the job they were hired for, but they did not understand security. The strong focus on safety permeated the culture, and in some control rooms, the passwords were written on the blackboard next to the control console. As he had found out six months later, these passwords were not changed regularly.

He had left his work at the platform, with enough money to at least begin his formal education at university. It was then that he had finally started to implement the attack. He guessed it could be seen as quite sophisticated from one perspective, because he had had to use several weaknesses in the system to leapfrog into the console he needed to control. On the other hand, he had not used any special types of attacks but he rather exploited a culture of disregard for security, one small step at a time, and together these steps had added up to quite an assault. Running an oil platform is a complex operation with several organizational entities that need to co-operate and communicate. By chance, his university had served as an expert node. In the event of a problem, experts could come here instead of going to the actual platform or the remote facilities in Great Britain. This node served as the entry into the system and a way to bypass the outermost layers of protection; the university network was, after all, recognized as an authorized peer. He had installed malicious software at the node, which then had been downloaded to the platform the first time the platform host had setup a connection to the university node. Tailored exactly to the local environment, the malicious software then reprogrammed the old router and leapfrogged onto the critical network where it had tried the old password to connect to the console. He actually still felt a bit of shameful pride that he had managed to implement these steps without having any direct link back to where he could manually interact with the malicious code to instruct it what to do. The software was instructed to only report on an IRC channel when it had reached certain goals. The last communication had been when the console had been infected. After that there had been no further communication from the program and he assumed he had failed. Unfortunately, he was proven wrong a couple of hours later.

The announcement of the large oil spill was broken on the radio. Soon television crews were at the scene and the large, contaminated area spreading over the ocean could be tracked online. The next day he could read in the morning paper that the safety systems on the platform had failed to stop the flow of thousands of tons of oil into the ocean. The reason for this failure, according to the paper, was that the computers on the platform had been infected by malicious code blocking the function of some critical safety systems. He had felt frustrated by the lack of detailed information. His program was not supposed to do any such thing, but there was no way he could investigate further. Instead, he joined the millions of others watching the oil get transported by the wind to the coast over the next couple of days. Then he joined the volunteers in the clean-up crew. For every bird he could not save, he felt a stab in his heart. His only comfort was the fact that fortunately none of the crew had been seriously hurt in the accident.

He looked at the beach again. The authorities had never been able to find the culprit. Little did they know that it was him, the director appointed to evaluate the long-term effects of the oil spill. He had thought about turning himself in, but it had seemed so pointless. Better that he actually did something concrete to atone for his crime instead of just wasting everyone's time and money. Yes, he nodded to himself. This October 5^{th} was actually just a day like any other. Some wondrous steps had been taken at this date, such as the first steps in the space adventure but also some horrendous accidents had happened on this date. For him personally, however, the day would always mark the death of his youth.

16.3 Consequences

There have been several oil spills, among them the Gulf War oil spill, Ixtoc I spill in the Gulf of Mexico, the Amoco Cadiz spill outside the coast of France, and the Exxon Valdez spill in the Gulf of Alaska. The consequences of large oil spills can affect a region disastrously from an economic standpoint but also be a catastrophe from an ecological point of view. Some environments are especially sensitive, and oil spills can destroy the intricate balance between the species. For that reason, a smaller spill in one environment might be more severe than a larger spill elsewhere. An editorial in New York Times [145] lists the consequences of the Exxon Valdez accident. 2000 km of shoreline was affected and one of the richest fishing grounds in the US was damaged; only some of the species of fish found before the accident made a recovery. The cleanup cost more than 500 million Euros and many workers lost their livelihood.

16.4 Related threats

Issues related to the *use of COTS components* (Threat #28) and *retrofitting security to legacy systems* (Threat #20) are fundamental to this scenario. Some of the systems are simply vulnerable to "normal" malicious code, and the antagonist uses this fact to download his code onto the offshore platform. The antagonist then uses his detailed knowledge of the system (*the insider threat* – Threat #13) for his next step in the attack. As *safety takes priority over security* (Threat #25) in many industrial domains, we emphasized the non-existing password policies at the offshore platform in the text. Other examples could also have been mentioned. As already discussed, the domain is complex, and issues discussed in *unforeseen cascading effects* (Threat #9), *threats to system maintainability and verifiability* (Threat #14), and *threats due to scale* (Threat #2) also work for the benefit of the antagonist. Even though not explicitly mentioned, the human factor probably played a role in this scenario, and better designed *user interfaces* (Threat #12) might have alerted the operators in time to the failure of the safety system. Finally, we would like to point out that the antagonist could never be brought to justice. Legacy systems, where security has not permeated the design, seldom have the necessary sophistication for allowing advanced forensic analysis.

16.5 Could it happen?

By taking advantage of knowledge known to any insider, as well as some luck, the antagonist managed to achieve his goals. The scenario is plausible, although an actual attack may exploit other types of vulnerabilities. For example, the article [83] reports that a dissatisfied former employee hacked into the system at Pacific Energy Resources, so that they lost control of their telemetry system. The employee had worked on remote telemetry and leak detection. No oil leak or environmental harm was caused in this particular case.

The case study of the attack against the Maroochy Water Services [2] is also important. In this case, the antagonist was able to connect to the system remotely at least 46 times before being discovered. It was easy for him to break into the system [15], it was difficult to distinguish between malfunctions and malicious activities, and it was very difficult to track his actions because normal security mechanisms were lacking in the system.

In [99], it is reported that the segregation between the IT network and the network used for monitoring critical applications is often inadequate. Control engineers with security experience are still rare but will play a vital role in the future. Over the years there have also been a number of accidents involving oil spills from offshore platforms [5].

According to experts within our working group, this scenario will be even more plausible in the future. The trend of increasing efficiency through using greater automation, remote operations, and reducing the level of crew offshore leads to ever more complex organizational structures relying on computer systems that are inherently more vulnerable to malfunctions and malicious attacks.

In our discussions with experts, there was an acknowledgment of the need to have control system engineers with a better understanding of security issues. Many formerly-closed systems are opened for remote access, sometimes without an adequate understanding of the risks involved. In our threat report, we have listed problems with wireless networks but any remote access poses risks. Furthermore, many of these safety systems are built with COTS components (meaning that they are still vulnerable to malware that is developed elsewhere) but they cannot be updated as easily as more traditional systems. Further research into building more resilient systems will make the above scenario less likely to occur. For example, some of these systems might be less prone to changes than traditional systems and thus certain security mechanisms that have proven difficult to adapt to desktop systems and commercial servers, such as anomaly-based methods [92] or a verified boot sequence [10], might be better suited in this domain. Systems are often necessarily complex, with many organizations actively cooperating with many types of hardware and software. An improved understanding of heterogeneous and complex systems as well as how security errors are introduced and propagate within the system leads to more secure systems. On top of that, the provision of some kind of security metric would make such security verifiable. Finally, better-designed user interfaces for control engineers would further reduce the likelihood of the above scenario. For example, fine-grained security monitoring techniques that scale across complex distinct networks would make the above scenario less likely.

Chapter 17

Scenario: Fabrication Take-over

17.1 Overview

This scenario illustrates how a motivated adversary with access to significant financial resources can leverage technology to cause global mayhem. In this scenario, the attacker, instead of carrying out "traditional" attacks against specific software or hardware components, manipulates the design of hardware to support the attacks. To manipulate the design of hardware for this purpose, the attacker must be, at least partially, involved in the fabrication process. The attacker may require significant financial resources to host and maintain a rogue fabrication facility, train people that will seek employment at an existing fabrication facility and act as insiders, carrying out the attack, or bribe possibly disgruntled employees. In each case the attacker may need significant resources, time to train the right people or locate and hire the right individual to prepare the attack. Nevertheless, once the attacker has succeeded to create the rogue hardware devices, they then have enough flexibility to massively orchestrate sophisticated attacks, since the malicious hardware has potentially reached thousands (or even millions) of end devices.

17.2 Scenario

The procedure hadn't changed in eight years. All units are transferred to Sector 7, where the controller chips for the wireless card interfaces are attached to each network device. It takes less than 80 hours for one million devices to be prepared and verified. The whole process is very reliable; no more than 3 in 500,000 units manifest errors. Strangely enough, this time Sector 7 was not responsible for attaching the controllers to the network cards. The instructions were quite clear, very specific and sent confidentially to the director of the plant. All network devices should be transferred to a new plant located 5km away in the same industrial zone where the main facility is located. Upon seeing the confidential letter, the director made the necessary arrangements for the procedure of unit-transfer to start at once. There was enough pressure lately for the whole company to release the new line of network interfaces, and the director didn't want to delay the procedure any more than absolutely necessary.

The new plant was built very recently, and this was the first massive fabrication it had to deliver. One million controllers in less than 48 hours, outperforming the older plant in the main facility. After 48 hours of non-stop work, every device was ready to ship to market. The new plant had successfully completed the task and delivered all network cards with their controllers attached.

Twelve months later...

The news were reporting strange series of incidents daily. Hundreds of users had their hard disks completely wiped out, suddenly, without leaving any clue as to what the reason was. The first few incidents were classified as hardware failures, but day-by-day the increase of strangely wiped out hard disks started to paint a different picture. A few additional details surfaced. All wiped-out hard disks were installed in laptops, but not of a specific brand. All major IT companies that sell laptops had received such reports. The affected people didn't seem to belong to a specific user class. Some of them were computer experts, some of them were not. Some of them were using Microsoft Windows, some of them Linux and others Mac OS X. Even OpenBSD users were being affected! It was hard for someone to claim that this was caused due to a virus. Only two facts were common in all cases. All users had their hard disk wiped out while they were connected to the Internet using a wireless connection, and while they were in the US (that is their laptop had an IP address that was geo-located in the United States).

After a few weeks, the reported incidents numbered in the thousands. Some of the laptops belonged to employees of the US government and US military. Some others were part of large installations of medical equipment and experiments in Biology and Physics. The strange virus had attacked thousands of laptops without any discrimination, apart from the fact that all laptops were wirelessly connected with a US IP address. The US government ordered an extensive investigation. Experts investigated all incidents, case by case, to conclude that all laptops had wireless controllers produced by one of three different manufacturers. All three companies responsible for producing those three controllers were ordered to recall and replace all of them (even if the equipped laptops were not affected). The financial losses are still not estimated, but unofficial guesses place them in the order of millions of dollars. The source of the problem as well as the motivation has yet to be identified. No charges have been brought to anyone and the case is still considered open.

17.3 Consequences

On the one hand, such an incident causes a significant financial loss. Devices need to be replaced, and a lot of data is lost. In addition, there is a loss of confidence in the reliability of computer systems in general, and the manufacturer of the "faulty" network cards in particular. While in this scenario, the attackers did not directly draw a benefit from the attacks, it could still be part of a terrorist plot that causes chaos in the United States (which was obviously targeted in this attack). Moreover, one could also image that instead of wiping out hard disks, data could have been stolen.

17.4 Related threats

The main threat described in this scenario is related to the problem of *malicious hardware* (Threat #27). However, there are also threats that manifest because of the complexity of the manufacturing processes and *threats due to scale* (Threat #2). Also, typically, the main purpose of tampering with the fabrication process is to introduce *hidden functionality* (Threat #15) into the hardware that is later distributed. Depending on the goal of the performed manipulations, a number of other threats will apply as well.

17.5 Could it happen?

The threat is clearly realistic. In fact, we encountered some first manifestations during the project life time. For instance, The US National Counterintelligence Executive, Joel Brenner, in 2008 announced that an organized crime group succeeded into tampering with commercial credit card readers which were shipped to and installed in retail stores around Europe. These devices are believed to have been in use for at least nine months before they were discovered and the fraud amounts to millions of euros.

A more technical description of some of the technical requirements to launch hardware attacks appears in [70]. As technology evolves and as the model that drives technology evolution becomes more complex, this kind of attack is more likely to happen. Nowadays, the hardware fabrication process is divided into multiple steps. There are different companies responsible for the construction of specific parts of a device. A modern laptop carries hardware parts that are constructed by tens of different companies, and we expect this number to increase in the near future as laptops start incorporating a multitude of new network interfaces and sensors. Controlling the entire fabrication process of a computer system is considered hard, if not impossible, and the probability of insiders that alter this procedure is high. Although the attacker needs significant resources to perform the attack, the results, if the attack succeeds, can be severe.

http://news.softpedia.com/news/Hundreds-of-Tampered-Chip-and-Pin-Devices-Spread-i
shtml

Chapter 18

Scenario: The Politician's Phone

18.1 Overview

This scenario describes how attackers compromise a fictional politician's smart phone. First, the attackers use the hacked phone simply to eavesdrop on confidential conversations. The information thus gleaned is used to obtain unlawful advantage in tender applications for large construction projects. Next, the attackers raise the stakes by using the phone to obtain privacy-sensitive and embarrassing information about the politician (the Under Secretary of Housing and Planning in a fictional country). They use the information to blackmail the politician, in order to influence the way in which the politician uses his powers. Finally, as the situation deteriorates, and the attackers' demands become increasingly bold, the politician feels he cannot comply any longer, and he refuses to play along. At that point, the attackers decide to damage the politician's reputation. This is done by planting compromising information that soon makes the politician's position untenable and force him to resign.

18.2 Scenario

18.2.1 The epilogue first

We will start our scenario with the end. The politician has been defamed, and he is pressed to resign. Imagine a report like the following in a newspaper:

> In a development late last night that could have wide repercussions for the political situation in Acountry, Mr. A. Non, Under Secretary of Housing and Planning, has been arrested and charged with trafficking child pornography. In a statement to the press, Mrs. X, a spokeswoman for the police explained that while the police investigations are still ongoing, "substantial amounts of very explicit material were found on the politician's mobile phone, much of which involves very young children."

> While the Rt. Hon. Andrew Non declined to comment, sources close to the accused say that he claims to "have no idea how the material came to be on his phone."
>
> Prime minister Y., of the same party, appeared shocked, but says he does not want to speculate and will await the results of the police investigations. The opposition parties, however, are calling for the accused to resign his post of Under Secretary pending the investigations.
>
> Starting out as an energetic young official, popularly known as *Mr. Internet* for his smart use of new communication technology during the election campaign, Andrew Non has in recent years sometimes been considered "too friendly" with construction companies allegedly linked with organized crime. While he was never formally charged with any wrongdoings, several MPs have called it remarkable that the same companies landed most of the large infrastructure construction projects in the capital city. Already controversial, the current trouble, therefore, seems almost certain to spell an end to Mr. Non's political career.

Not long after that, more damaging evidence appears. The politician's GPS tracker shows that he visited the red light district on at least nine occasions, while call records on his phone show that he regularly called the number of an expensive escort service. Confronted with this information, the politician denies to have made any such phone calls, but a spokeswoman for the prosecutor issues a statement saying that call records in several mobile phone providers confirm the telephone calls.

The politician loses his job and his reputation, and it is likely that he suffers greatly in his personal life also. In our scenario, the politician was the victim of a new high-tech wave of organized crime. In the remainder of this chapter, we will discuss what happened and how it could occur in reality.

18.2.2 What happened

The young politician, a keen user of the Internet, is known for keeping in touch with friends, colleagues, and the press, by means of his smart phone - a sleek new model, more advanced then, but not very different from the earlier iPhones, Androids, and Nokias. He uses his phone to Twitter, IM, email, browse, and maintain his presence on various social networks. And one day, he stumbles onto the wrong web site ...

This is no accident. The politician is a victim of a targeted attack. Knowing the model of the phone and the politician's general interests, a sophisticated group of hackers manages to compromise a social network site that is frequented by the politician. On some of the pages, likely to be visited by the politician, they place malicious content, which immediately compromises his phone's browser by means of a "drive-by download" attack.

Full control over the phone's browser gives the attackers a lot of power. For a while, they harvest information about the politician's browsing activities. But the compromise is fragile. As soon as the politician reboots the phone, the browser comes up "clean" and the attackers lose control over it. Fortunately for them, they manage to compromise the browser on multiple occasions, but it is important that they find a more permanent hack.

By exploiting a vulnerability in the phone's kernel, they manage to get access to the most privileged code running in the phone. Using these privileges, the attackers manage to modify the configuration files (or even to overwrite the phone's image) in such a way that each time it boots, it automatically starts with the attackers' code. To hide any trace of the compromise, the attackers change the system and the system utilities in such a way that the malicious code they run never shows up as a process and that the data generated by the malicious code never shows up in the file system as visible by the users. Such techniques are standard in rootkits and do not cost the attackers much effort.

Now that they have penetrated deeper into the politician's phone, the opportunities for abuse become wider. The attackers snoop on the politician's emails, messages, and keystrokes. With a bit more coding, they gain control over the CPU and the microphone, allowing them to prevent the phone from going to sleep and switching on the microphone (and perhaps the camera) at will.

Thus, the attackers are able to listen in on confidential meetings of the cabinet, and particularly, negotiation meetings concerning large housing and infrastructure projects. The attackers sell the information to two friendly construction companies who use the information to gain unlawful competitive advantage in bidding for contracts.

18.2.3 The plot thickens

We could end the scenario here. However, let us instead shift gears and assume that the politician is not altogether without blemish - as is sometimes the case with politicians. We could think of many possible blemishes, but let us assume a fairly mild one: the Under Secretary is unfaithful to his wife. One day, the attackers switch on the microphone (and perhaps the camera), just as the Right Honourable consumes his extra-marital affair.

The attackers decide to blackmail the unfortunate Under Secretary. The politician is highly embarrassed, and while he does not know how the villains were able to make recordings right there in the hotel room, he does know that he does not want his wife or his colleagues to find out. As the first demands made by the blackmailers are fairly mild, he plays along. He never sees the bad guys. Orders are given by phone.

As time progresses, the blackmailers make bolder demands. When the politician protests, they replay for him recordings of earlier conversations in which he agrees to help the criminals. Before long, he is partly "owned" by the criminal organization, who now influences a substantial number of important decisions about real estate and infrastructure – to the advantage of organized crime.

The politician realizes that this cannot continue and one day, he draws the line and threatens to call the police. The attackers feel threatened and decide to defame the politician in order to get rid of him. They use his phone to download and spread pornography and to make phone calls and send messages to escort services and phone numbers associated with organized crime. They also fake GPS trails that show the politician frequently visiting the red light district.

The calls, downloads, and messages are all genuine, and they were all made from the politician's phone. The attackers make sure to post illegal content from the phone in ways that can be easily traced by law enforcement agencies. This is where we began the scenario: the politician is charged and on his phone the police finds a large amount of compromising material. Even if it is not enough to convict the politician, it is enough to finish his career and ruin his private life.

18.3 Consequences

The immediate consequences of the scenario sketched above include financial loss, damage to reputation, blackmail, and corruption of politics. However, the threat is more general than this specific scenario. In general, a compromised mobile device with different types of sensor means that attackers can eavesdrop on individuals in almost all spheres, including very confidential and private ones. The result may be used directly, or for blackmail purposes. Moreover, the scenario shows that control over devices allows attackers to defame people by planting "evidence." Even if the evidence does not hold up in a court of law, it is generally sufficient to tarnish the victim's reputation.

18.4 Related threats

The scenario directly maps on the threats related to smart phones and *mobile device malware* (Threat #4), loss of *privacy through ubiquitous sensors* (Threat #11), and *false sensor data* (Threat #10). Smart phones are like small PCs in the range of applications and also in their vulnerabilities to attacks. However, our use of smart phones is still rooted in our habits stemming from traditional phone usage. In other words, while the attacks may be familiar, they may lead to subtly different attack vectors for attackers (spyware in the bed- or meeting room). Moreover, a compromised device can easily be used to plant very real-looking false evidence (including calls to escort services, or bogus location data). Finally, the attackers performed a *targeted attack* (Threat #26) that was directly aiming at the Under Secretary.

18.5 Could it happen?

It is more than likely that smart phones will be vulnerable enough to allow attackers to compromise the phone *completely*. Indeed, this is already the case with current phones. The amount of code on these phones increases in scope and complexity and thus in the number of bugs and exploitable vulnerabilities [48, 116, 118]. Vulnerabilities in the past have allowed attackers to use Bluetooth to completely take over mobile phones of various vendors, such as the Nokia 6310, the Sony Ericsson T68, and the Motorola v80. The process, known as bluebugging, exploited a bug in Bluetooth implementations [82]. While these are older phones, more recent models, such as the Apple iPhone have also shown to be susceptible to remote exploits [98, 112, 116], including the drive-by downloads that were used in the above scenario [107].

Once the attackers control a process, it is generally fairly simple to become root and thus control the entire phone, including the kernel. While it is slightly harder to hack the kernel to make it boot the malicious code, or even change the boot image on a phone, this is certainly possible on several current phones (including the Android G1). Rootkit technology that helps attackers evade detection is also common in many operating systems [143]. A common trick to evade detection is to change the system call table to execute the attacker's code instead of the original code. The attacker's code then makes sure that no trace of the attackers' files or activities when the user lists all files in a directory or all processes running on the machine.

Moreover, with the full power of the phone, sensors can be switched on and off at will. Similar things have happened to webcams attached to PCs in the past. The main difference here is that we carry our phones around, wherever we go (including meeting rooms and bedrooms). In other words, the smart phone can be turned into the most effective and personal spies imaginable.

Worse, while smart phones are like small PCs in terms of processing capacity, range of applications, and vulnerability to attacks, we generally cannot simply apply the same security measures as are currently available on desktop PCs. Phones and PCs may have started to look similar in applications, but they still differ in significant aspects, most notably power and physical location. These two aspects matter when it comes to security. Unlike normal PCs, smartphones run on battery power, which is an extremely scarce resource. For instance, one of the main points of criticism against Apple's iPhone 3G concerned its short battery life [100]. Vendors work extremely hard to produce efficient code for such devices, because every cycle consumes power, and every Joule is precious.

As a consequence, many of the security solutions that work for desktop PCs may not be directly portable to smartphones. Anti-virus file scanners [96], reliable intrusion detection techniques [86], and other well-known techniques all consume battery power and may not be applicable to the phones.

The remaining question is whether the scenario itself is realistic. Observe that the politician was victim of a targeted attack: the attackers specifically aimed for *his* phone. Currently, we often see attacks that spread as widely as possible (botnets, for instance). However, if the victim is sufficiently important - such as an Under Secretary - it may well be worth the attackers' while to target specifically that individual. Whether the victim is a politician, a CEO, or someone else with influence is immaterial. It is certain that the threat is taken seriously at certain levels, witness US president Barack Obama's famous struggle to keep his Blackberry smartphone, after he was told this was not possible due to security concerns. It is not unlikely that attackers will aim for influential figures just below the absolute top tier - say the Under Secretary, rather than the Secretary or Prime Minister.

In our opinion, the first part of the scenario is very plausible indeed. In fact, an attack on a switch in the phone system in Greece has allowed unknown attackers to eavesdrop on conversations of the Prime Minister and several other important individuals. The main difference with the scenario sketched above is that, in our scenario, the attackers compromise an individual's phone rather than a (potentially better secured) telephony switch.

Clearly, the second half of the scenario depends greatly on the politician. If he has secrets that a compromised phone can reveal and that make him vulnerable to blackmail, this is certainly not implausible. But even if he does not, the compromised phone can be used to defame the politician (or businessman) by making calls, sending out emails, storing pictures, and so on. In our view, the security aspects of next generation smart phones cannot be overstated.

It is unlikely that in the near future phones will be made sufficiently safe to trust them completely in all spheres. Since smart phones are expected to be the main interface to the Internet in the future, it is essential that even if we cannot prevent attacks altogether, we will be able to check reliably whether or not they have been compromised *a posteriori*. Additionally, it is important that users are aware of the security implications and that smart phones are barred from certain spheres - including important meetings. While the success of user education in the past has been mixed at best, it is relatively simple for organisations to implement policies to prevent participants from bringing mobile phones into important meetings.

http://spectrum.ieee.org/telecom/security/the-athens-affair

Chapter 19

Scenario: Mass Blackmailing through Social Networks

19.1 Overview

This scenario outlines a series of possible threats that can arise by using a social network. The key properties in play are (a) the scale of a social network and (b) the implicit trust among typical users. Both properties, (a) and (b), transform a social network into an ideal platform for distributing content and information to a significant user base. First, as far as property (a) is concerned, the intended content can reach millions of user. Second, as far as property (b) is concerned, users trust the content, since it is frequently suggested by their contacts (one-hop friends). Moreover, users also upload content related to their private life. So far, modern social networks cannot guarantee that a user's content is not accessible by third, possibly malicious, parties. An adversary can collect specific data related to a user and then use this data to launch targeted social engineering attacks against the victim.

19.2 Scenario

Alice was reading an article in the local newspaper about the extent to which social networks have penetrated the townfolks' lives. The article was narrating how many well-known stores in the little town had created a group page on Facebook. The story was also reporting how a significant number of citizens that took part in a survey reported that they had a user account in MySpace, Facebook, or Twitter. A few words about Alice: She is a computer owner and she is occasionally using a computer at her work. She was never a big fan of computers, and her major interest was in linguistics. It was the first time that Alice read an extensive article about social networks. She was really amazed and curious. It only took her a few minutes to create her brand new Facebook account and join the social network. Despite her not being a computer expert, the site was extremely easy for her to use. She managed to easily find the search service and actually locate some old classmates from the time she was studying in college; they had lost touch for over a decade. Alice was impressed by the power of this "web site" and could now see why people had joined as massively as the article in the newspaper was reporting.

The next morning, Alice visited Facebook immediately after clearing-out her inbox. Her typical web surfing routine was to visit Gmail, some news sites, and maybe a short look at Amazon, but that was it. She was now logged into Facebook, watching her news feed. Her eyes dropped to the invitation box. She was pleasantly surprised to see an invitation from Marco, her Spanish roommate in college, to install a cool application that ... can predict her future! Alice, without any second thoughts, immediately installed the application and waited for the prediction. The application predicted that Alice is going to have two lovely twins! She laughed, brushed it aside, and continued "killing some time" on the social network.

By the end of the month, Alice had become extremely familiar with the social network and appreciated its utility. Spending a few hours in Facebook every day became part of her daily routine. She knew how to upload her photos, she was occasionally chatting with old friends, and, from time to time, she installed applications suggested by her contacts. Some of her favorite applications included "Are you lucky today?" (an application that randomly predicts your daily luck), "Your daily quiz" (a quiz generator) and "Garden-Maniac" (a strategy game where the user has to take care of a virtual garden). Alice, quickly, started to investigate the various privacy options of Facebook and became familiar with them, too. She configured everything in order to be sure that no information will leak to third parties, such as her photos or data from her account's description. Although Alice had no such concerns before, having all these privacy settings as an option in her personal preferences made her feel even more confident about her choice to upload parts of her personal life to the social site.

It was a Friday evening when a message in Alice's Facebook inbox changed everything. The message was originating from an unknown Facebook user. It contained a few selected thumbnails of Alice's personal pictures. Alice had uploaded these pictures to Facebook, but she had made sure, through the privacy settings, that they were only visible to her friends. A team of cyber-hooligans, as they were calling themselves, had, somehow, gotten hold of all of Alice's photos and they were going to publish them all over the Internet. In order to not do so, they were asking her to deposit 1,000 Euros to a bank account controlled by the cyber-hooligans. Alice had not uploaded any photo, which if it were to become public, would put her in a difficult position. Nevertheless, Alice felt an immediate panic and fear about the social network. It wasn't about the money or about the photos, but the fact that anybody could penetrate into her private life, disrupt it and even blackmail her. Alice decided to reset her Facebook profile. This was her last interaction with Facebook and the online social network world.

Later the same month, the news reported that a massive blackmailing campaign was held in one of the most successful social networks, Facebook.com. The attackers had spread a malicious application, which pretended to generate funny quizes.

In reality, while the user was solving the quiz, the application was collecting all the user's photos and sending them to the attackers' server. The application managed to reach more than 10 million installations in less than a month. When the attackers succeeded in collecting a significant number of pictures, they started sending blackmail messages to the pictures' owners, asking them for money (ranging form a few Euros to thousands of Euros, depending on the number of pictures they had collected for each victim) in order to not publish the photos on the Internet. As soon as disrupted users reported the incident to Facebook, the company removed the malicious application. A Facebook spokesman announced that they could not estimate the losses from users that had agreed and deposited the demanded ransom. They had received various complaints by users, but there was no clear plan for a possible refund, since Facebook itself did not publish the malicious application.

19.3 Consequences

The consequences in this scenario are relatively minor for individual users. They might feel threatened and abandon the social network, and they might suffer some financial loss by paying the demanded sums. The social network provider loses trust from its user base. However, the underlying problem is that private and sensitive data for a large number of users is unprotected and can be abused by attackers in various ways. The scale of the opportunities for attackers is the main concern here.

19.4 Related threats

The scenario describes *threats related to social networks* (Threat #5). The information stolen in this case is used to make profit and support the *underground economy* (Threat #3). In addition, collected information could be leveraged to carry out *targeted attacks* (Threat #26).

19.5 Could it happen?

Researchers have pointed out in the past that malicious applications can exist in popular social networks that provide APIs for building dynamic content [12, 79, 151]. The malicious application in this scenario is technically similar to the ones developed in [12]. Additionally, researchers have tried to build technologies that provide the users more fine-grained control over the amount of personal information that may leak over the social platform [138]. So far, these technologies have not been deployed, but the fact that the research community is actively involved towards this direction supports the case that information leakage is a major issue in social networks. In this scenario, there are no casualties, and only disturbance and maybe some financial losses. However, similar cases in the past have lead to death (through suicide) [93].

Chapter 20

Scenario: Attack against an Electric Power Station

20.1 Overview

There is a major trend to implement ICT in critical systems. This direction is taken for years and will continue, since new technologies bring compelling positives to these systems: They gain additional functionality, some services can be accessed and controlled remotely, and large systems are connected and seem more easily manageable. This significant deployment efforts, however, come also at the price of security problems, and the price could be very high. In this scenario, we illustrate how a (organized) group of motivated attackers access the main functionality of an electric power station and wreck havoc, with possibly severe consequences, to achieve their malicious goals.

20.2 Scenario

The world-wide economic crisis is a fact, and the recession is going deep. The leaders of the small Party for National Progress (PNP) lost in the last parliamentary elections. Now, they have interest in a further destabilization of the economy considering the overall political situation in the country, because they have lost the opportunities to legalize their underground business. They have a lot of money, but lost their power and are eager to regain it. In addition, the winter is coming, and along with that, the next gas crisis is quite probable.

As a result of the crisis, there are many unemployed people. Tom was working as computer and network expert at the biggest electric power station (EPS) in the country. He was laid-off and, as a result, he is disgruntled and disappointed. Before the crisis, the EPS has always been using leased lines from the national telecommunications company (N-telco) and special radio channels for communications with the National Control and Distribution Center (NCDC), responsible for the National Power Distribution Grid (NPDG) management and monitoring. Now, due to economic considerations, these lines are replaced by VPN and VoIP respectively, and the special radio channels are used only for back-up and emergency purposes. Meanwhile, in the last 2-3 years, the N-telco accomplished a rapid transition to NGN technology.

To exploit these favorable circumstances, the leaders of PNP need somebody inside the N-telco to help carry out their malicious plan. They found John, an expert in the N-telco, who is sympathetic to their political and ideological cause and is prone to become a voluntary agent in the furtherance of that cause, along with the opportunities to earn good money.

The plan for the cyber attack is not simple because it comprises many people and facilities (bearing resemblance to the organized crime) and all the actors involved want to go out unnoticed and without any prosecution. Moreover, they intend to reuse the cyber threat scenario with some modification multiple times if the first attempt succeeds, attaining the main goal – economic havoc and destabilization.

The leaders of PNP promised a lot of money and gave some amount in advance to Tom to try to cause an accident in the EPS. The pressing financial motivation is overshadowing motivations related to personal fame and reputation, and Tom joins the attackers.

As Tom is knowledgeable about the architecture and functions of all systems and networks at the EPS, he knows that there is more than one monitoring station above the supervisory level, transmitting regularly data to the Ministry of Ecology for surveillance. These monitoring stations have a continuous connection to the Internet and usually get read-only data from Level 1 (data acquisition, information processing and monitoring, and control). Moreover, he is aware that some of these stations are also linked, via VPN, with the NCDC. The VPN is considered secure, and when the communication session originates from the VPN, the monitoring stations can be given access directly to Level 0 (main process control functionality area), again for read-only data (but in real time). This access is realized through the OPC protocol, even though the main process control uses another specialized and vendor-specific protocol. Furthermore, Tom knows that for the smooth operation of OPC, the firewall between Level 1 and Level 0 is switched off for the time of the communication session.

Tom has all the information he needs about the critical processes and controls in the EPS, but after his lay-off, security measures were strengthened. Thus, he could not gain access to the monitoring stations anymore. At this point, John enters the game.

John is perfectly aware of the VPN (connecting EPS and NCDC) and waits for an opportune moment to assign Tom full access privilege level to the VPN.

The NPDG has a specialized autonomous and independent system for automatic dispatching and management of power distribution. For our scenario we have forced a bit the fiction, but as the trends are "all-going-IP," this threat is emerging and very probable in the near future.

Such a moment could be the short time interval just before the regular patching of some of the VPN servers, which, on the other hand, cannot be arbitrary due to the uninterruptability of VPN's operation. Because the VPN was built using VPN access servers from different vendors, John does not have to wait long for this moment to come. The main goal of choosing such a moment is to cover the traces, and thus, to hide the source of attack, and at the same time, to imitate plausible deliberate intrusion from the outside. Already having access to the VPN, Tom initiates an OPC session, which automatically switches the firewall off. This allows him to reach one of the working stations at Level 0. This working station directly monitors the operation of the programmable logic controllers (PLCs), and, at the same time, serves as operators' interface for non-automatic and emergency control. Here, Tom installs a small malware program, some kind of time bomb, which after some time delay will destroy the normal functioning of the PLCs. After installing the malware, he erases all the logs in the system at the entering point (at the VPN). The choice of this type of cyber attack with time delay additionally impedes the tracing of the attack, because the time of intrusion into the VPN and the time of the actual attack to the PLCs do not coincide.

20.3 Consequences

If the cyber attack is successful, it is possible to provoke, in extremely short time (seconds-to-minutes), a very serious plant incident – an emergency shut down of one or more power units, which in turn is capable of disturbing the energy system of the entire country. To worsen the situation, the gas crisis and heavy winter conditions can lead to peak load in electric heating and other higher power consumption, thus overburdening the NPDG. Since the targeted EPS is the biggest one, given the bad circumstances, cascading effects are quite possible in the NPDG with unforeseen economic losses.

20.4 Related threats

The previously-described scenario shows the risks that are due to *the insider threat* (Threat #13), whose malicious activities are considerably facilitated by the increased vulnerabilities of the N-telco services after the deployment of *NGN technologies* (Threat #21). Moreover, it illustrates the problem of the use of *COTS components and systems* (such as OPC) in SCADA systems (Threat #28), as well as the common problem that *safety takes priority over security* (Threat #25) in such settings. Also, the scenario touches on the problem of *threats due to scale* (Threat #2) and *unforeseen cascading effects* (Threat #9) because of the increasing interdependences between different CIs through NGNs, and the tight connections between power stations. Of course, also the problem of *advanced malware* (and time bombs - Threat #18) plays a role in carrying out the attack successfully.

There are also threats that are not cyber, such as the world economic crisis, the gas crisis and heavy winter conditions, as well as the politic situation represented by the Party for National Progress (PNP). Critical infrastructures are exposed to such a diversity of risks.

20.5 Could it happen?

The SCADA part of the scenario was shown to an expert in Industrial Automation Process-Control Systems, and it was considered realistic. As for the N-telco of NGN type, some specific vulnerabilities of the NGNs are described in [39] and [127] and discussed in Section 10.4.

A systematic study of *32 fundamental vulnerabilities in NGN* and five scenarios, most of them concerning critical systems, can be found in [111]. In addition, cascading effects are not unlikely, as some recent examples show [26].

We are just at the beginning of exploitation of industrial control systems' vulnerabilities (see [144, 11]). Therefore, more research and activities for improving the functions and security of SCADA and OPC are needed.

Aside from the insider threat, which is very hard to find and neutralize, the security of NGNs is still an open problem. As is pointed out in [39], a security program is necessary and its success depends on: "identity management; linking privileges to identities; automated adaptive security mechanisms for networks that seek out malware and unusual activities."

A NGN Scenario Threat Profile Matrix is described, which details anticipated threats for each user class within the context of an NGN National Security/Emergency Preparedness scenario.

Chapter 21

Scenario: Attack on a Water Power Station

21.1 Overview

This scenario describes how a relatively easy-to-launch attack against a water power station can be performed. The attackers exploit known vulnerabilities and bad security practices. No matter how advanced the implemented technologies are, there still exist unprotected areas in critical systems that have to be secured.

21.2 Scenario

The Plin water power station is located 50km away from the nearest town. Peter has always liked to work there. Up in the mountains, life looks simple. He takes a deep breath of fresh air while approaching the entrance. The two security guards just nod when they see him. They are busy preparing their coffee for the night shift. "What a boring job," Peter thought. "To sit all night long in front of the monitors, watching the same familiar images from the surveillance cameras. Good that sometimes they walk around to move their feet. Thank God, my job is not so dull!"

Entering the control room, he leaves all worries behind him. The quiet noise of computers and printers fills him with calm. Nick is already there. He informs him about the latest communication with the center – nothing to worry about. Monitors show the usual figures and curves. Peter sits in his chair and starts to check the connection. Recently, a wireless network was installed, which connects to the corporate wired network through wireless bridges. This technology is still new to him, and he wants to be sure that everything the IT guy taught him is running okay.

It was decided to install a wireless network because it was considered too expensive to have cables connecting the water power station with the corporate center, and it is difficult to maintain such an infrastructure. To save resources and time, a wireless network connection was implemented. Peter was excited, he liked the opportunity to use this modern technology. Of course, there are still implementation problems and many new things to learn. All these annoying accounts, passwords ... Why just not sit and work!

It is getting dark. A car is climbing slowly the steep winding road to the water station. The power station is situated far from any residential area, and there is little chance that the car will be spotted. The car stops at a small flat area with clear visibility to the station. The driver turns off the lights. A man exits from the car and installs a (powerful) direct antenna. Sitting in the car, the two men switch on their laptops and quickly see the wireless network appear on the screen. Even if the area were crowded with wireless connections, it would not be difficult to recognize the name "Plin station." They start searching the network. The wireless network is based on the 802.11 standard. The authentication is unidirectional – only from the control room devices/users to the network. No authentication of the network is done by local users. It should not be difficult to disguise their activity with the latest toolkit downloaded from the Internet.

It is fun to test this new attack tools. Should they work as they are supposed to, the power will be on their side. Causing a blackout just with a laptop and antenna ... One of the men shivered just with the thought of the amount of money they could get from selling their skills. It will not actually be so easy to do it, but with a little luck they can make it.

The attack proceeds as follows:

- War driving to locate wireless devices
- Sniffing to identify the MAC addresses of the operators' computers in the control room of the station
- A missing password (disabled during the testing process) helps the attackers to access one of the control room computers
- Attackers establish connection of their wireless devices to the corporate network
- To the control room the rogue network looks like the corporate network
- Man-in-the Middle (MITM) attack
- Attackers take over the direct control of the station

While studying the new capabilities of the wireless connection, Peter sees a strange line appearing on the screen. He checks with the instructions the IT guy gave him; no such warning should be there. "I'll ask him tomorrow morning," he thought. "Why bothering the guy with trifle!" He checks again the connection, and it looks perfectly operating.

Minutes later, the turbine is shutting down.

21.3 Consequences

As in the previous scenario, access to the OPC server and the SCADA system can be used to shutdown the system, but here it is done through the wireless network. The MITM attack disguises the intrusion and slows down the prompt reaction of the center that could have prevented the shutdown and the possible cascading effects. Failure in one power station can often propagate quickly to others, causing blackouts due to the lost electrical grid balance.

21.4 Related threats

A main threat in this scenario stems from the fact that *wireless communication* (Threat #8) is used without the necessary security precautions. This allows attackers to access networks inside buildings and plants without the need to enter the premises. Moreover, as is often the case with industrial setups, *safety takes priority over security* (Threat #25). Specifically, in the scenario, control experts perform functional testing but do not complete the security procedure. Also, forgotten or omitted passwords are common. There is also the problem of *retrofitting security to legacy systems* (Threat #20). Here, the standard 802.11i was not implemented from the design phase of the network. Of course, given the nature of the attacked plant, *unforeseen cascading effects* (Threat #9) in the form of series of blackouts are very probable.

21.5 Could it happen?

Some examples of successful attacks against the critical infrastructure already exist (e.g., [147, 135]). Although they are not wide-spread yet, there are signs of a growing interest from the cyber crime community of attacks directed towards the critical infrastructure [113]. The misconception that SCADA systems require specialized knowledge, and thus, are difficult for network intruders to access and control, is no longer appropriate. Information describing SCADA system operations, as well as OPC architecture, is available. There are standards and recommendations describing how to connect systems together and how to issue controls. Moreover, there are documentation and toolkits for developing software for SCADA environments.

On the other hand, critical infrastructures have always been targets for terrorists and with the implementation of "open" technologies they are getting more exposed to cyber-terrorism, which is organized and well equipped.

The real-world examples of attacks show that up to now most of the attacks to critical systems are isolated acts of disgruntled employees, single hackers or are side effects of attacks directed to other connected systems. This is probably due to the fact that the most attractive targets for a cyber-terrorist attack are well protected and that there are other easier and more profitable ways of achieving one's adversarial goals. For a highly motivated antagonist, however, no obstacle is insurmountable.

The control community is getting aware of the cyber risks to which their systems and networks are exposed. New standards are developed and approved to build security into critical systems. For example, there are new and more secure OPC specifications [117] and SCADA standards. Recommendations and good practices are widely published (see, for example, [1]). A lot of work is going on and there are still open areas for research in this domain.

Bibliography

[1] 21 Steps to Improve Cyber Security of SCADA Networks. President's critical infrastructure protection board and department of energy report. http://www.oe.netl.doe.gov/docs/prepare/21stepsbooklet.pdf, 2002.

[2] M. D. Abrams and J. Weiss. Malicious control system cyber security attack case study – Maroochy water services, Australia, Aug. 2008. http://csrc.nist.gov/groups/SMA/fisma/ics/documents/Maroochy-Water-Services-Case-Study_briefing.pdf.

[3] Advancedtrading.com. The Real Story of Trading Software Espionage. http://advancedtrading.com/algorithms/showArticle.jhtml?articleID=218401501.

[4] J. A. Afilias, P. Savola, and G. Neville-Neil. Deprecation of type 0 routing headers in ipv6. IETF, 2007.

[5] AFP. Large oil spill near North Sea oil platform: Norway, Dec. 12, 2007. http://afp.google.com/article/ALeqM5gKH21ZKRRYqBvlHS1Q-1BK51G_Fg.

[6] Aite Group. Algorithmic Trading: Hype or Reality? http://www.aitegroup.com/reports/20050328.php.

[7] P. Akritidis, W. Y. Chin, V. T. Lam, S. Sidiroglou, and K. G. Anagnostakis. Proximity breeds danger: emerging threats in metro-area wireless networks. In *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16, Berkeley, CA, USA, 2007. USENIX Association.

[8] D. Anderson, C. Fleizach, S. Savage, and G. Voelker. Spamscatter: Characterizing internet scam hosting infrastructure. In *Usenix Security Symposium*, 2007.

[9] R. J. Anderson and M. G. Kuhn. Low cost attacks on tamper resistant devices. In *Proceedings of the 5th International Workshop on Security Protocols*, pages 125–136, London, UK, 1998. Springer-Verlag.

[10] W. Arbaugh, A. Keromytis, D. Farber, and J. Smith. Automated recovery in a secure bootstrap process. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS'98)*, pages 155–167, 1998.

[11] ASL Communicating by design, ModernUtility Management. A critical situation. http://www.modernutilitymanagement.com/article-page.php?contentid=6555&issueid=222, Nov., 24 2008.

[12] E. Athanasopoulos, A. Makridakis, S. Antonatos, D. Antoniades, S. Ioannidis, K. G. Anagnostakis, and E. P. Markatos. Antisocial networks: Turning a social network into a botnet. In *ISC '08: Proceedings of the 11th international conference on Information Security*, pages 146–160, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] P. Baecher, T. Holz, M. Koetter, and G. Wicherski. Know Your Enemy: Tracking Botnets, 2007.

[14] D. Balzarotti, G. Banks, M. Cova, V. Felmetsger, R. Kemmerer, W. Robertson, F. Valeur, and G. Vigna. Are Your Votes Really Counted? Testing the Security of Real-world Electronic Voting Systems. In *International Symposium on Software Testing and Analysis (ISSTA)*, 2008.

[15] G. Barker. Cyber terrorism a mouse-click away. Internet, July 8, 2002. http://www.theage.com.au/articles/2002/07/07/1025667089019.html.

[16] U. Bayer, A. Moser, C. Kruegel, and E. Kirda. Dynamic analysis of malicious code. *Journal in Computer Virology*, 2(1):67–77, 2006.

[17] M. Becher and F. Freiling. Towards dynamic malware analysis to increase mobile device security. In *Sicherheit 2008: Sicherheit, Schutz und Zuverlässigkeit. Konferenzband der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 2.-4. April 2008 im Saarbrücker Schloss*, pages 423–433, 2008.

[18] J. Berra. Emerson first to offer WirelessHART automation products. `http://www.controlglobal.com/industrynews/2008/082.html`, 2008.

[19] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. In *20th International World Wide Web Conference*, Madrid, Spain, April 2009.

[20] A. Bose and K. G. Shin. On mobile viruses exploiting messaging and bluetooth services. *Securecomm and Workshops, 2006*, pages 1–10, 28 2006-Sept. 1 2006.

[21] A. Caglayan, M. Toothaker, D. Drapeau, D. Burke, and G. Eaton. Real-time detection of fast flux service networks. *Conference For Homeland Security, 2009. CATCH '09. Cybersecurity Applications & Technology*, pages 285–292, March 2009.

[22] H. Chan and A. Perrig. Security and privacy in sensor networks. *IEEE Computer*, 36(10):103–105, 2003.

[23] J. Cheng, S. H. Wong, H. Yang, and S. Lu. Smartsiren: virus detection and alert for smartphones. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 258–271, New York, NY, USA, 2007. ACM.

[24] Cisco Systems Inc. Annual Security Report. www.cisco.com/go/securityreport, 2008.

[25] ComputerWeekly.com. Israeli Trojan espionage writers extradited for trial. `http://www.computerweekly.com/Articles/2006/02/01/213977/israeli-trojan-espionage-writers-extradited-for-trial.htm`.

[26] Cunnecticut Post. Bad weather blamed in blackout for 60M in Brazil. `http://www.connpost.com/business/ci_13760627`, Nov., 11 2009.

[27] D. Dagon, G. Gu, C. Lee, and W. Lee. A taxonomy of botnet structures. In *Annual Computer Security Applications Conference (ACSAC)*, 2007.

[28] M. Davis. Smartgrid device security: Adventures in a new medium, July 2009. `http://www.blackhat.com/presentations/bh-usa-09/MDAVIS/BHUSA09-Davis-AMI-SLIDES.pdf`.

[29] dear cots. DEAR-COTS project homepage. `http://dear-cots.di.fc.ul.pt`, 2001.

[30] F-Secure. E-spionage. `http://www.f-secure.com/weblog/archives/00001424.html`.

[31] F-Secure. F-secure computer virus information pages: Cardtrap.a. `http://www.f-secure.com/v-descs/cardtrap_a.shtml`.

[32] F-Secure. F-secure computer virus information pages: Commwarrior.a. `http://www.f-secure.com/v-descs/commwarrior.shtml`.

[33] Facebook. `http://www.facebook.com`, 2009.

[34] Fake Facebook pages spin web of deceit. http://www.nature.com/news/2009/090423/full/news.2009.398.html, 2009.

[35] Federal Bureau of Investigation. SPEAR PHISHERS Angling to Steal Your Financial Info. http://www.fbi.gov/page2/april09/spearphishing_040109.html, Apr 2009.

[36] E. Felten. NJ Election Day: Voting Machine Status. http://www.freedom-to-tinker.com/blog/felten/nj-election-day-voting-machine-status, 2008.

[37] C. Fleizach, M. Liljenstam, P. Johansson, G. M. Voelker, and A. Méhes. Can you infect me now? malware propagation in mobile phone networks. In *Proceedings of The 5th ACM Workshop on Recurring Malcode (WORM 2007)*, 2007.

[38] A. Folkerts, G. Portokalidis, and H. Bos. Multi-tier intrusion detection by means of replayable virtual machines. Technical Report IR-CS-47, Vrije Universiteit Amsterdam, August 2008.

[39] D. P. M. Fonash. Cybersecurity & Communications (CS&C) Overview, Technology Trends, & Challenges. http://events.sifma.org/uploadedFiles/Events/2008/BCP/Fonash%20presentation.pdf, Oct. 2008. Homeland Security.

[40] FORWARD Consortium. Managing emerging threats in ICT infrastructures: Threat report (deliverable D2.1.x), 2009. http://www.ict-forward.eu/media/publications/forward-d2.1.x.pdf.

[41] J. Franklin, V. Paxson, A. Perrig, and S. Savage. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. In *ACM Conference on Computer and Communication Security (CCS)*, 2007.

[42] F. D. Garcia, G. Koning Gans, R. Muijrers, P. Rossum, R. Verdult, R. W. Schreur, and B. Jacobs. Dismantling mifare classic. In *ESORICS '08: Proceedings of the 13th European Symposium on Research in Computer Security*, pages 97–114, Berlin, Heidelberg, 2008. Springer-Verlag.

[43] T. Goodspeed. Extracting keys from second generation Zigbee chips, July 2009. http://www.blackhat.com/presentations/bh-usa-09/GOODSPEED/BHUSA09-Goodspeed-ZigbeeChips-PAPER.pdf.

[44] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. H. Kang, and D. Dagon. Peer-to-Peer Botnets: Overview and Case Study. In *1st Workshop on Hot Topics in Understanding Botnets*, April 2007.

[45] S. Guha, K. Tang, and P. Francis. NOYB: privacy in online social networks. In *Proceedings of the first workshop on Online social networks*, pages 49–54. ACM New York, NY, USA, 2008.

[46] S. Gundersson. Global IP V.6 Statistics - Measuring the Current State of IPv6 for Ordinary Users. Technical report, RIPE 57, 2008.

[47] M. Hansen, M. Hansen, J. Möller, T. Rohwer, C. Tolkmit, and H. Waack. Developing a legally compliant reachability management system as a countermeasure against spit. In *Proceedings of Third Annual VoIP Security Workshop, Berlin, Germany*, 2006.

[48] L. Hatton. Reexamining the fault density component size connection. *Software, IEEE*, 14(2):89–97, 1997.

[49] T. Holz, M. Engelberth, and F. Freiling. Learning More About the Underground Economy : A Case-Study of Keyloggers and Dropzones. Technical report, University of Mannheim, 2008.

[50] T. Holz, M. Engelberth, and F. Freiling. Learning More About the Underground Economy: A Case-Study of Keyloggers and Dropzones. In *European Symposium on Research in Computer Security (ESORICS)*, 2009.

[51] T. Holz, C. Gorecki, and F. Freiling. Detection and Mitigation of Fast-Flux Service Networks. In *Network and Distributed System Security Symposium (NDSS)*, 2008.

[52] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. C. Freiling. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[53] The Honeynet Project. Know Your Enemy: Fast-Flux Service Networks., July 2007.

[54] Honeynet Project and Research Alliance. Know your Enemy: Tracking Botnets. `http://www.honeynet.org/papers/bots/`, 2008.

[55] G. Huston. The IPv4 Address Report. `http://www.potaroo.net/tools/ipv4/`, 2008.

[56] IEEE. IEEE Standard for Information technology. `http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=35824&arnumber=1700009&punumber=11161`, 2006.

[57] IPv6 Related Specifications. `http://www.ipv6.org/specs.html`, 2006.

[58] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, 2007.

[59] M. Jakobsson. Modeling and Preventing Phishing Attacks. `http://www.informatics.indiana.edu/markus/papers/phishing_jakobsson.pdf`, 2005.

[60] M. Jakobsson and S. Stamm. Invasive browser sniffing and countermeasures. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 523–532, New York, NY, USA, 2006. ACM.

[61] H. James. The Teredo Protocol: Tunneling Past Network Security and Other Security Implications. Symantec, 2006.

[62] Y. Jing. Fast Worm Propagation in IPv6 Networks. `http://www.cs.virginia.edu/~jy8y/publications/cs85104.pdf`, 2006.

[63] J. John, A. Moshchuk, S. Gribble, and A. Krishnamurthy. Studying Spamming Botnets Using Botlab. In *6th Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[64] A. Juels, R. L. Rivest, and M. Szydlo. The blocker tag: selective blocking of rfid tags for consumer privacy. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 103–111, New York, NY, USA, 2003. ACM.

[65] H. Kagan. Interview about wireless devices adoption in the industry and the future trends. Frost & Sullivan, Nov. 2008. `http://www.teknikogviden.dk`.

[66] C. Karlberger, G. Bayler, C. Kruegel, and E. Kirda. Exploiting Redundancy in Natural Language to Penetrate Bayesian Spam Filters. In *First USENIX Workshop on Offensive Technologies (WOOT '07), Boston, MA*, August 2007.

[67] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *In First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.

[68] Kaspersky Lab. Kaspersky lab reports a new malicious program for mobile phones that steals money from mobile accounts. `http://www.kaspersky.com/news?id=207575728`, January 2009.

[69] M. Keeney. Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors Executive Summary. `https://treas.gov/usss/ntac/its_report_050516_es.pdf`, May 2005.

[70] S. T. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang, and Y. Zhou. Designing and implementing malicious hardware. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–8, Berkeley, CA, USA, 2008. USENIX Association.

[71] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, pages 104–113, London, UK, 1996. Springer-Verlag.

[72] M. Konte, N. Feamster, and J. Jung. Dynamics of online scam hosting infrastructure. In *PAM*, pages 219–228, 2009.

[73] New Koobface Worm Variant Spreads Across Facebook, Myspace, Hi5 And Other Social Networks. `http://cyberinsecure.com/new-koobface-worm-variant-spreads-across-facebook-myspace-hi5-and-other-social-networks/`, 2009.

[74] B. Krebs. Shadowy Russian Firm Seen as Conduit for Cybercrime. `http://www.washingtonpost.com/wp-dyn/content/article/2007/10/12/AR2007101202461.html`, 2007.

[75] Krisberedskapsmyndigheten. Omvärldsexempel 2005. Published by Krisberedskapsmyndigheten, Sweden, 2005. KBM:s dnr:0280/2005.

[76] Krisberedskapsmyndigheten. En sammanfattning av rapporten: faller en – faller då alla? ISBN: 978-91-85797-24-0. Published by Krisberedskapsmyndigheten, Sweden, 2007. KBM:s dnr:0021/2007.

[77] P. Kumaraguru, S. Sheng, A. Acquisti, L. Cranor, and J. Hong. Lessons from a real world evaluation of anti-phishing training. In *eCrime Researchers Summit, 2008*, pages 1–12, Oct. 2008.

[78] G. Kurtz. Operation Aurora Hit Google, Others. `http://siblog.mcafee.com/cto/operation-œaurora-hit-google-others/`, 2010.

[79] V. T. Lam, S. Antonatos, P. Akritidis, and K. G. Anagnostakis. Puppetnets: misusing web browsers as a distributed attack infrastructure. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 221–234, New York, NY, USA, 2006. ACM.

[80] L. Laursen. Fake facebook pages spin web of deceit. *Nature*, 458, 2009.

[81] F. Leder and T. Werner. Know Your Enemy: Containing Conficker, 2009.

[82] G. Legg. The bluejacking, bluesnarfing, bluebugging blues: Bluetooth faces perception of vulnerability. `http://www.wirelessnetdesignline.com/showArticle.jhtml?articleID=192200279`, Apr. 2005.

[83] J. Leyden. IT contractor charged over US oil rig hack: Roughneck cracker charges. Internet, Mar. 19, 2009. `http://www.theregister.co.uk/2009/03/19/oil_rig_hack_charges/`.

[84] Z. Li and D. Lee. Detecting and filtering instant messaging spam-a global and personalized approach. In *1st IEEE ICNP Workshop on Secure Network Protocols, 2005.(NPSec)*, pages 19–24, 2005.

[85] LinkedIn. `http://www.linkedin.com`, 2008.

[86] M. Costa, J. Crowcroft, M. Castro, A Rowstron, L. Zhou, L. Zhang and P. Barham. Vigilante: End-to-end containment of internet worms. In *SOSP*, Brighton, UK, October 2005.

[87] M. T. Hoske and I. McPherson. Industrial Wireless Implementation Guide. Control Engineering, 8/1/2008, Aug. 2008. `http://www.controleng.com/article/CA6584939.html`.

[88] M. Mannan and P. van Oorschot. On instant messaging worms, analysis and countermeasures. In *Proceedings of the 2005 ACM workshop on Rapid malcode*, pages 2–11. ACM New York, NY, USA, 2005.

[89] M. Manzo, T. Roosta, and S. Sastry. Time synchronization attacks in sensor networks. In *SASN'05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 107–116, New York, NY, USA, 2005. ACM.

[90] K. Masica. Recommended Practices Guide For Securing ZigBee Wireless Networks in Process Control System Environments, Draft. http://csrp.inl.gov/Documents/Wireless%20802.11i%20Rec%20Practice.pdf, April 2007.

[91] K. Masica. Securing WLANs using 802.11i, Draft Recommended Practice. http://csrp.inl.gov/Documents/Wireless%20802.11i%20Rec%20Practice.pdf, Feb. 2007.

[92] R. Maxion and K. Tan. Anomaly detection in embedded systems. *IEEE Transactions on Computers*, pages 108–120, 2002.

[93] Suicide of megan meier. http://en.wikipedia.org/wiki/Suicide_of_Megan_Meier.

[94] MeinVerzeichnis – MeinVZ. http://www.meinvz.net/, 2008.

[95] Microsoft.com. Spear phishing: Highly targeted phishing scams. http://www.microsoft.com/protect/yourself/phishing/spear.mspx, Jul 2008.

[96] Y. Miretskiy, A. Das, C. P. Wright, and E. Zadok. Avfs: an on-access anti-virus file system. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 6–6, Berkeley, CA, USA, 2004. USENIX Association.

[97] D. Moore, G. Voelker, and S. Savage. Inferring internet denial of service activity. In *Usenix Security Symposium*, 2001.

[98] H. Moore. Cracking the iphone (part 1). Available at http://blog.metasploit.com/2007/10/cracking-iphone-part-1.html, October 2007.

[99] A. Moses. 'sinister' integral energy virus outbreak a threat to power grid. Internet, Oct. 1, 2009. http://goo.gl/uNqO.

[100] W. Mossberg. Newer, faster, cheaper iphone 3G. Wall Street Journal, July 2008.

[101] S. Moyer and N. Hamiel. Satan is on My Friends List: Attacking Social Networks. http://www.blackhat.com/html/bh-usa-08/bh-usa-08-archive.html, 2008.

[102] C. Mulliner. Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones. In *Proceedings of the 1st International Workshop on Sensor Security (IWSS) at ARES*, Fukuoka, Japan, March 2009.

[103] C. Mulliner and G. Vigna. Vulnerability Analysis of MMS User Agents. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Miami, FL, December 2006.

[104] C. Mulliner, G. Vigna, D. Dagon, and W. Lee. Using labeling to prevent cross-service attacks against smart phones. In *Detection of Intrusions and Malware & Vulnerability Assessment, Third International Conference, DIMVA 2006, Berlin, Germany, July 13-14, 2006, Proceedings*, pages 91–108, 2006.

[105] MySpace. http://www.myspace.com, 2009.

[106] New MySpace and Facebook Worm Target Social Networks. http://www.darknet.org.uk/2008/08/new-myspace-and-facebook-worm-target-social-networks, 2008.

[107] R. Naraine. Google Android vulnerable to drive-by browser exploit. *http://blogs.zdnet.com/security/?p=2067*, October 2008.

[108] J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, pages 24–31, 2008.

[109] M. Neely. My Facebook Nightmare. http://infolution.com.au/?p=112, Jan 2009.

[110] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 259–268, New York, NY, USA, 2004. ACM Press.

[111] Next Generation Networks Task Force, Appendices. Appendix G: Systematic Assessment of NGN Vulnerabilities, Appendix H: NGN Threat Analysis. `http://www.ncs.gov/nstac/reports/2006/NSTAC%20Next%20Generation%20Networks%20Task%20Force%20Report%20-%20Appendices.pdf`, March, 28 2006.

[112] Niacin and Dre. The iphone / itouch tif exploit is now officially released. Available at `http://toc2rta.com/?q=node/23`, October 2007.

[113] NIST. Guide to industrial control systems (ICS) security, Sept. 2008. SP800-82, `http://csrc.nist.gov/publications/drafts/800-82/draft_sp800-82-fpd.pdf`.

[114] T. Noonan and E. Archuleta. The National Infrastructure Advisory Council's final report and recommendations on the insider threat to critical infrastructures. `http://www.dhs.gov/xlibrary/assets/niac/niac_insider_threat_to_critical_infrastructures_study.pdf`, April, 8 2008.

[115] O. Nordstrom and C. Davrolis. Beware of bgp attacks. In *ACM SIGCOMM Computer Communications Review*, 2004.

[116] oCERT. CVE-2009-0475: #2009-002 opencore insufficient boundary checking during mp3 decoding. `http://www.ocert.org/advisories/ocert-2009-002.html`, January 2009.

[117] OPC Foundation. OPC Specifications. `http://opcfoundation.org/`.

[118] A. Ozment and S. E. Schechter. Milk or wine: Does software security improve with age? In *15th USENIX Security Symposium*, Vancouver, BC., July 2006.

[119] P. Welander. "Securing Legacy Control Systems". `http://www.controleng.com/article/307540-Securing_Legacy_Control_Systems.php`, Jan., 7 2009.

[120] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi. Fluxor: Detecting and monitoring fast-flux service networks. In D. Zamboni, editor, *DIMVA*, volume 5137 of *Lecture Notes in Computer Science*, pages 186–206. Springer, 2008.

[121] B. D. Payne, M. Carbone, M. Sharif, and W. Lee. Lares: An Architecture for Secure Active Monitoring Using Virtualization. In *IEEE Symposium on Security and Privacy*, 2008.

[122] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *Commun. ACM*, 47(6):53–57, 2004.

[123] B. Philippe and E. Arnaud. IPv6 Routing Header Security. CANSECWEST, 2007.

[124] Playstation Home. `http://www.playstationhome.com`, 2009.

[125] P. Porras, H. Saidi, and V. Yegneswaran. A Multi-perspective Analysis of the Storm (Peacomm) Worm. Technical report, Computer Science Laboratory, SRI International, 2007.

[126] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The Ghost In The Browser. In *First Workshop on Hot Topics in Understanding Botnets (HotBots)*, 2007.

[127] H.-Y. Y. Rack-Hyun Kim, Jae-Hoon Jang. An Efficient IP Traceback mechanism for the NGN based on IPv6 Protocol. `http://jwis2009.nsysu.edu.tw/location/paper/An%20Efficient%20IP%20Traceback%20mechanism%20for%20the%20NGN%20based%20on%20IPv6%20Protocol.pdf`, 2008.

[128] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *ACM SIGCOMM*, 2006.

[129] A. Rassiysky. Evolution of data networks of BTC. Cisco Expo, Oct. 2008. http://www.cisco.com/web/BG/expo/presentations/BTK_Evolution_of_Data_Networks_of_BTC.pdf.

[130] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. Is your cat infected with a computer virus? In *PERCOM '06: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 169–179, Washington, DC, USA, 2006. IEEE Computer Society.

[131] SANS Institute. Malware infection that began with windshield fliers. http://isc.sans.org/diary.html?storyid=5797, 2009.

[132] P. Savola and C. Patel. Security considerations for 6to4. IETF, 2004.

[133] T. Schaberreiter, C. Wieser, I. Sánchez, J. Riekki, and J. Röning. An enumeration of rfid related threats. In *UBICOMM '08: Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 381–389, Washington, DC, USA, 2008. IEEE Computer Society.

[134] Second Life. http://www.secondlife.com, 2009.

[135] SecurityFocus. Slammer worm crashed Ohio nuke plant network. http://www.securityfocus.com/news/6767, Aug. 2003.

[136] ShadowServer: DigitalNinja. RBN 'Rizing' - Abdallah Internet Hizmetleri. http://www.shadowserver.org/wiki/uploads/Information/RBN_Rizing.pdf, 2008.

[137] ShadowServer: Pheh. RBN As a Business Network - Clarifying the guesswork of Criminal Activity. http://www.shadowserver.org/wiki/uploads/Information/RBN-AS40989.pdf, 2008.

[138] K. Singh, S. Bhola, and W. Lee. xbook: Redesigning privacy control in social networking platforms. In *Proceedings of the 18th Usenix Security Symposium*, August 2009.

[139] J. Smith. Security: Stolen Facebook Accounts Being Used to Phish for Money from Friends. http://www.insidefacebook.com/2009/01/21/, Jan 2009.

[140] Spear phishing: Highly targeted phishing scams. http://www.microsoft.com/protect/yourself/phishing/spear.mspx, 2006.

[141] StudiVerzeichnis – StudiVZ. http://www.studivz.net, 2008.

[142] Symantec. Palm.phage.dropper. http://www.symantec.com/security_response/writeup.jsp?docid=2000-121918-4538-99.

[143] Symantec. Windows rootkit overview. *White Paper, http://www.symantec.com/avcenter/reference/windows.rootkit.overview.pdf*, 2005.

[144] The iDefense Security Intelligence Team. 2009 Cyber Threats and Trends. http://www.verisign.com/idefense/information-center/resources/whitepaper-idefense-2009trends.pdf, Dec.,12 2008.

[145] The New York Times Editorial. Lessons of the Exxon Valdez. Internet, Mar. 22, 2009. http://www.nytimes.com/2009/03/23/opinion/23mon1.html.

[146] The President's National Security Telecommunications Advisory Committee. Next Generation Networks Task Force, Report. http://www.ncs.gov/nstac/reports/2006/NSTAC%20Next%20Generation%20Networks%20Task%20Force%20Report%20-%20Appendices.pdf, March, 28 2006.

[147] The Register. Hacker jailed for revenge sewage attacks. http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/, Oct. 2001.

[148] R. Thomas and J. Martin. The Underground Economy: Priceless. *Usenix ;login*, 31(6), 2006.

[149] D. R. Thompson, N. Chaudhry, and C. W. Thompson. RFID security threat model. In *Acxiom Laboratory for Applied Research (ALAR) Conf. on Applied Research in Information Technology*, Conway, Arkansas, March 2006.

[150] United States General Accounting Office. Report to the subcommittee on emerging threats, cybersecurity, and science and technology, committee on homeland security, house of representatives, June 2008. `http://www.gao.gov/new.items/d08607.pdf`.

[151] B. E. Ur and V. Ganapathy. Evaluating attack amplification in online social networks. In *W2SP'09: 2009 Web 2.0 Security and Privacy Workshop*, Oakland, California, May 2009.

[152] U.S. Department of Homeland Security, Science and Technology Directorate. National power grid simulation capability: Needs and issues. Internet. `http://www.anl.gov/ese/pdfs/PowerGridBrochure.pdf`, Dec. 9–10, 2008. National Power Grid Simulator Workshop, Argonne, Illinois.

[153] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, September 2008.

[154] C. J. Walter, N. Suri, and T. Monaghan. Evaluating COTS standards for design of dependable systems. Proc. of the 2000 Int. Conf. on Dependable Systems and Networks, 2000, pp. 87–96., 2000.

[155] Wikipedia. Next generation networking (NGN_all-IP), Dec. 2008. `http://en.wikipedia.org/wiki/Next_Generation_Networking`.

[156] Wired. Scan of Internet Uncovers Thousands of Vulnerable Embedded Devices. `http://www.wired.com/threatlevel/2009/10/vulnerable-devices/`, 2009.

[157] R. Wojtczuk. Adventures with a certain Xen vulnerability (in the PVFB backend). Technical report, Invisible Things Lab, 2008.

[158] Xing – Global Networking for Professionals. `http://www.xing.com`, 2008.

[159] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 267–278. ACM New York, NY, USA, 2006.

[160] G. Yuxi. IP Bearer Network for NGN. `http://wwwen.zte.com.cn/main/include/showemagazinearticle.jsp?articleId=9559&catalogId=12165`, 2005.

[161] K. Zetter. E-Vote Software Leaked Online. `http://www.wired.com/politics/onlinerights/news/2003/10/61014`, 2003.

[162] Q. Zheng, T. Liu, X. Guan, Y. Qu, and N. Wang. A new worm exploiting ipv4-ipv6 dual-stack networks. In *WORM '07: Proceedings of the 2007 ACM workshop on Recurring malcode*, 2007.

[163] T. Zhimeng, W. Bo, and W. Yinxing. Security Technologies for NGN. `http://wwwen.zte.com.cn/main/include/showemagazinearticle.jsp?articleId=11167&catalogId=12165`, 2008.