A COMPREHENSIVE THREAT ASSESSMENT FRAMEWORK FOR SECURING EMERGING TECHNOLOGIES

by

Ameya M Sanzgiri

December 6, 2013

A dissertation submitted to the Faculty of the Graduate School
of the University at Buffalo, State University of New York
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science and Engineering

Abstract

Wireless devices are becoming an integral part of the human environment and their seamless integration has created a range of new wireless sensor network architectures. Unfortunately the security of such networks often lags behind other advances and more often than not is developed only after the core systems and protocols have been standardized. This results in these security schemes having case-specific reactive attributes and being unable to anticipate any changes in the attacker's attack vector. Integrating security into the next generation computer applications core design is paramount as traditional "reactive" security operations on top of normal functionality will be an expensive and ineffective proposition. The primary focus of this dissertation is to develop a framework that assists in the formulation of proactive security schemes. A proactive security scheme aims at dissuading an adversary from attacking a system by increasing the cost of attack. Such schemes need to be integral to the design of the emerging technologies domain, so that protection against attacks, especially the stealthy and smart ones can be devised. However, to effectively design such schemes, one needs to understand the threats to a system as well as their effects on a system. Threat modeling in itself is a significant research challenge due to the lack of easy to understand techniques or methodologies. The problem of creating a framework which is generic enough for emerging systems and networked applications and can be easily adapted to provide a defender with appropriate attack vectors and risk analysis capabilities is considered. First a paradigm shift in threat modeling by incorporating the attackers perspective in the implementation of an attack and analyzing the various factors that

an attacker would have to consider in her attack is presented. Second, the identification of the avenues where the proposed framework can be used to increase the effectiveness of the modeling techniques is discussed. Although the framework can be used at any abstracted level, the dissertation focused on some of the most important avenues of attacks by studying the problem of identifying levels which present the most likelihood of risks. The levels discussed are (i) Architecture level - where the model is applied to the entire architecture, considering the specifics of the architecture and investigating threats to the architecture; (ii) Protocol level - where protocol (network) specifics and threats to the protocols are considered and (iii) Application level - where the threat model considers the application specifics, such as the purpose of the application and the unique features of the applications as well as the information from the architectural and protocol level threat modeling. The framework is applied to several existing as well as emerging real-world applications and open-ended attacks to identify and analyze the risks faced by these applications. The threat modeling approach considers epidemic theory to understand the degree of spread of malware using an online social network like Twitter, when one of the users is infected. Similarly, probabilistic modeling is used to understand the structure of a social network which would help in propagation of malware and concepts from complexity theory help in analyzing the cost of creating an attack to infect users in social networks. The risk verification is done via simulations as well as real world experiments. The aim of this research is to develop a framework, which will be a valuable aid in the creation of sound security schemes and risk analysis in the future.

"To my parents, who taught me the most important things in my life and to Euclid."

Acknowledgements

The writing of a dissertation is the highlight of any Ph.D. student's academic life. It is also the right time to express gratitude to everyone who have helped in this journey. This dissertation is the direct result of my advisor Dr. Shambhu Upadhyaya's terrific and patient guidance over the past few years. Dr.Upadhyaya was very generous with his time and was also very patient in dealing with me and my ideas, for which I shall always be grateful. His encouragement has been the necessary element which led me to do research (even outside my area of specialization) and complete this dissertation. He has been an incredible mentor and teacher, and the results in this dissertation are but a fraction of what I have learnt from him.

The debt I owe to the members of my dissertation committee is immense. I extend my sincere thanks to Dr. Chunming Qiao for his support on different parts of my research which go beyond this dissertation as well as for the various stimulating and interesting discussions we have had. Dr. Qiao has helped me understand the various subtleties of technical research and writing which I will always remember. I also want to thank Dr. Sheng Zhong for his advice, insights and discussions at the end of his class. Dr. Zhong's view on research is truly inspiring and I will always keep his anecdote on how the seemingly useless centuries old Euclidean algorithm resulted in today's PKI defining RSA algorithm, as a constant motivator for the rest of my life.

A significant phase of my life has been spent in Buffalo and I would like to thank several friends for making it enjoyable. My erstwhile colleagues, Dr. Raghuram Sudhaakar and Vel Prateesh Sankar have

been great to know and I have shared many memorable moments with them. Many thanks to the future doctorates - Aditya Wagh, Ananda Tirtha, Andrew Hughes and Ramanujam Sheshadri for sharing the joys and pains of a graduate student's life and for all the good times we have had.

My parents have always been a constant source of support and encouragement for me. They have always inspired me to aim higher and have provided me with confidence even when I found it lacking. No number of words can do justice towards the gratitude and love I feel towards them.

Last but not least, I want to thank God Almighty for His blessings and guidance.

Contents

1 Introduction **1**

1.1 Security Practices 2

1.2 Threat Modeling 4

1.2.1 Problems with Threat Modeling 4

1.2.1.1 Classes of Systems and Attacks 5

1.2.1.2 Disparity Between Attackers and Defenders . . . 6

1.2.1.3 Designer Involvemnet 7

1.2.1.4 Where All to Apply 8

1.3 Summary of Contributions 8

1.4 Roadmap 10

2 Background and Related Work **12**

2.1 Overview 12

2.2 Threat Modeling 12

2.2.1 Traditional Threat Modeling 13

2.3 Architecture 21

2.4 Protocol 22

2.5 Application 23

2.6 Summary 25

3 A Generic Framework for Threat Assessment **26**

3.1 Overview 26

3.2 A Generic Framework 27

3.2.1 Assumptions of the Framework 27

3.2.2 Modus Operandi of an Attack 28

3.2.2.1 Motivational Factors of an Attack 29
3.2.2.2 Motivation of an Attacker 29
3.2.2.3 Probability of Attack 29
3.2.2.4 Easier Alternative 30
3.2.2.5 Target Network Characteristics 30
3.2.2.6 Cost of Attack 30
3.3 Summary 34

4 Threat Modelling at the Architecture Level 35
4.1 Overview 35
4.2 Exploiting SSH on Mobile Devices 36
4.2.1 iOS Jailbreaking and SSH 36
4.2.2 iOS SSH Vulnerabilities 37
4.2.2.1 Remedies Against iOS SSH Vulnerabilities 38
4.2.3 Android Rooting and SSH 38
4.2.3.1 Android SSH Vulnerabilites 39
4.3 Smartphones as Aggressors/Victims 39
4.3.1 Attack Overviews 39
4.3.2 Synthesizing Attack Scenarios 40
4.4 Experimental Setup and System Model 41
4.4.1 SYN Attack Tool and Attack Procedures 42
4.5 Experiments and Results 43
4.5.1 Scenario 1 - Laptop as Aggressor 43
4.5.2 Scenario 2 - Smartphone as Aggressor 47
4.6 Summary 48

5 Protocol Level Threat Modeling 49
5.1 Overview 49
5.2 MAC Layer Jamming in WSN 49
5.2.1 Preliminiaries 50
5.2.1.1 Characterization of a Jamming Attack 50
5.2.1.2 Profiles of a Jammer 50
5.2.1.3 Severity of Jamming Attack 51
5.2.2 Effectiveness of Jamming Attack 52

5.2.3 Consideration of Jammer's Perspective 52
5.2.4 Attacker's Perspective and Concerns 53
5.3 Jamming Attack Risk Model 57
5.4 Summary 58

6 Application Level Threat Modeling 59
6.1 Overview 59
6.2 Preliminaries 60
6.2.1 Twitter User Model 60
6.2.2 Twitter Vulnerabilities 62
6.2.3 Attacks on Twitter 63
6.2.4 Attacks on Twitter 63
6.3 Analyzing the Impact of Malware Propagation using Twitter . . . 64
6.4 Results 69
6.5 Threat Model of Twitter for Spreading Malware 71
6.5.1 A Common Attack Methodology 71
6.5.1.1 Analysis of the attack 72
6.5.2 An Advanced Self-Propagating Attack 72
6.5.2.1 Analysis of the Advanced Attack 74
6.6 A Complex Indirect Attack 75
6.6.1 Analyzing the Complex Attack Scenario 77
6.6.1.1 Posting Malicious Links all the Time 77
6.6.1.2 Probabilistically Posting Links 78
6.7 Extension to Hash Tags 80
6.7.1 Analysis of Attack in the #-Tag Model 80
6.7.1.1 Miscreant Enters Trending #-Tag 81
6.7.1.2 Miscreant Creates Her Own Trending #-tag . . . 81
6.8 Cost Analysis and Discussions 82
6.8.1 Parameters of Interest to a Miscreant 82
6.8.2 Simulation Results 84
6.8.3 Cost Analysis 88
6.8.4 Discussion 90
6.8.4.1 Factors not considered 91

6.8.4.2 Lack of real-world experiments 92
6.9 Summary 93

7 Conclusion **94**

List of Publications **100**

References **114**

List of Figures

1.1 An overview of Security Practices 2
1.2 Perspectives of Attack and Defence 7

3.1 Steps involving an Attack 28
3.2 Relation between Factors affecting the Risk Assessment in our Model 33

4.1 Smartphone battery consumption under various scenarios 45
4.2 Effect of the size of Ping Packets on battery life when ping flooded for 20 minutes 45
4.3 Battery Drain Time for different ping packet sizes and corresponding packet loss percentages observed 46
4.4 Effect of ping packet on Network Characteristics 46
4.5 Laptop Consumption under Various Scenarios 47

5.1 Steps an attacker has to take for a Jamming Attack 55
5.2 Risk Model Applied to Jamming Attacks 56

6.1 Twitter Structure 61
6.2 How short URLs work in Twitter (numbers depict sequence of operation) 62
6.3 Timeline of attacks on Twitter 64
6.4 The spread of malware to other users/followers 67
6.5 Progression of Infection 69
6.6 Progression of Infection-II 70
6.7 Progression of Infection-III 70
6.8 Attack scenarios depicting the simple attack 73

6.9 The effect of the no. of trials on probability of malicious link being clicked 85
6.10 The probabilistic estimate of no. of susceptible for different depths 86
6.11 The effect of varying the no. of followers and probability of clicking on links on the number of susceptible users 87
6.12 Screenshot of Twitter simulator for illustration purposes only . . . 88
6.13 Comparison of theoretical and simulated probability of malicious links for various number of trials 89
6.14 Number of infected users for different click probabilities 90

7.1 A sybil scenario 96

List of Tables

6.1 Notation Summary 73
6.2 Cost Analysis Targeting 91
6.3 Cost Analysis Targeting Number of Susceptible 92

1

Introduction

"Security is a process, not a product."

– Bruce Schneier, *Crypto-Gram Newsletter*

Wireless technology has become ubiquitous in today's world and the trend is set to grow. This in turn has led to the inception and creation of a number of other support technologies and architectures. However, the advances made in these technologies often ignore aspects of security and security mechanisms are almost always added after the development of *core* systems and technologies. This leads to vulnerabilities which an adversary or miscreant can easily exploit. Integration of security as after-the-fact presents several research challenges since these technologies are diverse in terms of architectures, applications and implementations. Clearly, the basic step in integrating security features is is the formulation of a *threat model*. The definition of threat modeling that is used and is relevant to this dissertation is – "Threat modeling is the process of enumerating and risk-rating malicious agents, their attacks,and those attacks' possible impacts on a system's assets." [Steven 2010]. This dissertation focuses on conceptualizing a framework for threat modeling that is generic in nature, builds on lessons from older technologies and also takes into account the layers of the Open Systems Interconnection (OSI) stack.

The aim of this dissertation however is just not to conceptualize a framework, but rather to use this framework in the creation of proactive security schemes.

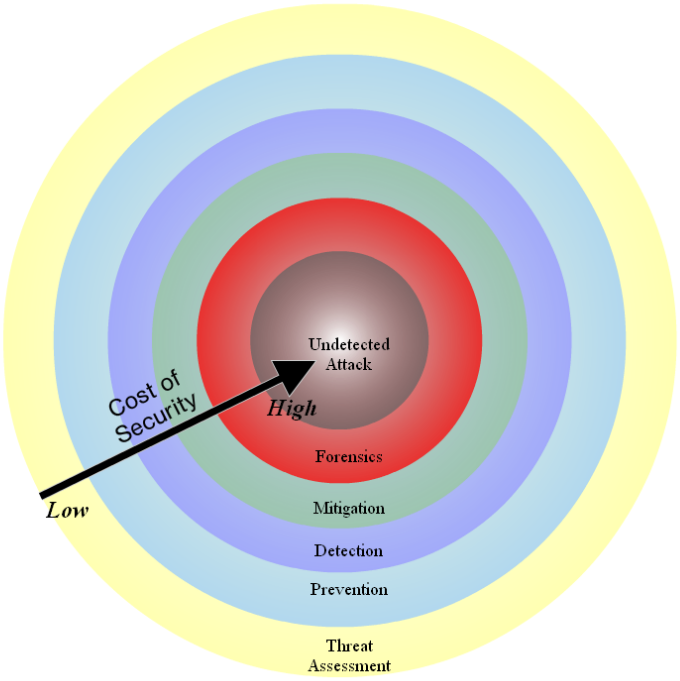

Figure 1.1: An overview of Security Practices

Proactive security schemes unlike traditional schemes are aimed at attack deterrence in the first place. While currently, these security schemes are not yet cost effective, this dissertation aims at providing good modeling techniques that examine attacks and the corresponding attack vectors that can be utilized in the future for the creation of light-weight proactive schemes.

1.1 Security Practices

Good security practices involve a multi-faceted layered approach. Despite of the fact that the eventual goal is the same at each of these stages, viz., to protect resources, the cost of security at these stages gradually vary as shown in Fig. 1.1. It must be noted that the boundaries between the stages are not necessarily distinct, in the sense that some of these stages may actually involve other stages as well.

1. Threat Assessment. Security needs vary from one organization to another and without a proper evaluation, it may be a case of bad investment and misplaced faith in security systems. Threat assessment begins by first identifying the likely targets which hackers may be interested in and then understanding the scope of the problem if any. Next, appropriate security devices can be installed in strategic locations to counter these threats.

2. Prevention. Preventive measures attempt to filter out possible attacks at various perimeters of the computational infrastructure using mandatory and discretionary access control mechanisms. These security measures are typically not concerned with the specifics of an attack. An example is a firewall which stops any suspicious network activity from entering or leaving an organization's network. Preventive security practices are known to be most effective in combating commonly occurring attacks termed as script-kiddies, however, they may not be adequate for more persistent and sophisticated attackers.

3. Detection. When access control is no longer adequate, it becomes necessary to understand the characteristic symptoms of an attack, and if the detection occurs early enough, then the attack can be preempted. Informally, the process of detecting attacks is called intrusion detection. Intrusion detection is essentially an event-driven decision-making process which operates on some form of input data such as a network packet stream or an audit log, and when some suspicious input is seen, then alerts are raised.

4. Mitigation. It is well-known that not all attacks can be prevented and some of them will be successful in spite of all the security measures that have been deployed. In such a scenario, the best that can be hoped for is to mitigate the damaging effects of an attack.

5. Forensics. Once an attacker has evaded all these security countermeasures, then the best that can be done is a postmortem analysis to isolate the damage and recover critical data to the extent possible. Forensics is also used to trace back an attacker's identity depending on whether the act was considered a criminal one.

An attacker who is able to penetrate every additional level of defense gets increasingly capable and dangerous. As mentioned before, the per-incident cost in the event that a particular level of security fails also increases dramatically. For example, forensics of a single security incident can take several computers and man-hours while a well-configured firewall alone can thwart several attack attempts.

The field of computer security is very vast and while there are several open problems to be solved, the focus of this dissertation is on the particular problem of threat assessment or threat modeling. Being the first step (and also at the least cost) to any good security process, one would assume that this step gets a lot of attention. However, threat modeling by itself is largely ignored since it is perceived to be "too costly", time-intensive and difficult to execute [Steven 2010]. This is largely due to the combination of the lack of formal techniques and the need for a thorough understanding of vulnerabilities of a system. In the following section we present an overview of threat modeling and the problems associated with threat modeling is presented.

1.2 Threat Modeling

The need for security necessitates the evaluation of the threats involved. As new technologies are developing, the security requirements and the measures to counter threats need to be constantly reviewed. The personal network devices are becoming more and more compact and their integration is getting seamless. The existence of threat has evolved the need for threat modeling. The systematic and comprehensive analysis of threats to the information system's confidentiality, integrity and availability needs to be done to ensure the security of the system. Threat modeling presents an inside-out view that provides more visibility than an outside-in black box or design assessment.

1.2.1 Problems with Threat Modeling

In the following section some of the reasons for threat modeling not being an integral process in practical security approaches are identified.

1.2.1.1 Classes of Systems and Attacks

A computer system (on a network) is basically a machine that is running an operating system and has multiple interfaces by which it can interact with the external world. These interfaces themselves interact with each other to provide a user with the means to access, modify and control data. However, access control alone cannot prevent malicious interactions or exploits. In the security domain, attackers typically exploit vulnerabilities resulting in unexpected or undesirable behavior of systems. Such steps taken by someone which are intentional and cause unauthorized or unintended behavior is defined as an attack. Attacks unlike faults and failures cannot be predicted via a design process, since it is a product of the mind set of an adversary or attacker.

One of the problems with threat modeling is that an attacker's approach to an attack varies with her intention or end result of the attack. For example, if an attacker intends to perform a Denial-of-Service (DoS) attack, the threat model would have to account for all avenues where such an attack could be possible (e.g., looking at the OSI stack for example); however, if the attacker intends to steal information, the avenues are going to be different (mostly the application layer). As recently depicted, there are viruses that affect Cyber-Physical Systems (CPS) that were previously thought of as fiction [Nicolas Falliere and Chien 2011], [Bencsáth et al. 2012] and [Goessl 2012]. Furthermore, there is a lack of information on how a host network (an already deployed network) would react to a combination of such attacks. With the number of attacks described in the literature the sheer computation of the combination of attacks would be a very difficult task in itself, not to mention the effort for modelling the threats for these combinations. Further using a specific technique to analyze threats for different technologies will be ineffective since the main purpose or core systems are diverse. To encourage threat modeling, a framework needs to be developed that while being generic is still flexible enough to account for the nuances or subtleties of technologies.

Problem Statement 1. Can we create a framework that helps model threats for different classes of attacks and for different systems?

1.2.1.2 Disparity Between Attackers and Defenders

Current security schemes are designed to protect against attacks as seen by the defender based on the limitations and vulnerabilities of her system. From a defender's perspective, the *entire* system is vulnerable to attacks and needs to be secured. Thus, the goal of a defender is to secure the complete system against all possible attacks. However, an attacker's perspective which is orthogonal to the defender's perspective, is to focus on a part of the system and attack. This difference in perspective is further highlighted in their individual goals where an attacker tries to find *one* flaw in the system and leverage it while the defender tries to defend her *entire* system by designing a security scheme. Currently the process of designing a security scheme relies heavily on Attack Graphs [Jha et al. 2002] and Attack Surfaces [Howard et al. 2003, Manadhata and Wing 2010] which are the two methods for formal assessment of risks. Attack surface is a conceptual tool used to increase the security of a software during its development. Attack graph is an abstraction that divulges the ways by which an attacker can leverage the vulnerability of a system to violate a security policy. It must be noted that in order to use the attack surface concept on a system, one has to know of all possible vulnerabilities and then optimize the available resources to cover the attack surface.

The inherent problem with the design of such schemes is that first, the defender does not have enough resources to completely secure her network.The countermeasures usually consider a single attack and are rarely feasible in terms of implementation complexity or cost to the network. Also, the defender is already at a disadvantage due to the fact that her perspective remains wide and vulnerable, while the attacker's perspective is more focused and specific. This methodology of designing security schemes has resulted in a performance as well as a feasibility gap of schemes in theory and practice which causes them to be reactive in nature. Figure 1.2 illustrates how a minimized attack surface created by a traditional security scheme which incorporates only the defender's perspective can still be viewed as a feasible attack surface from the attacker's perspective. This is different from the traditional approach as incorporating the attacker's perspective means the re-examination of some common assumptions with the goal of

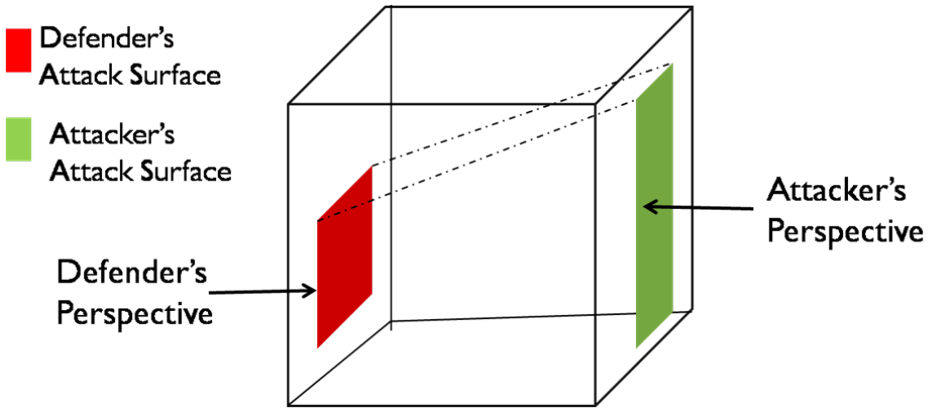

Figure 1.2: Perspectives of Attack and Defence

providing an effective yet practical outlook of the security of a system.

Problem Statement 2. Can we design a scheme to primarily reduce the gap and to minimize or eliminate the disadvantage of a defender by presenting a focused view of threats to the system in question?

1.2.1.3 Designer Involvemnet

Threat modeling is incomplete without involving people who design systems and are likely to be the ones who have an in depth understanding of their systems. However, as shown in [Steven 2010], designers perceive threat modeling as difficult since systems are complex and require collaboration implying that even threat modeling would require collaboration which many are unwilling to do. Another aspect is that for effective threat modeling, designers have to think like "attackers." This is analogous to asking "people who can cook, to think like a chef" [Shostack 2011]. While it is true that designers lack context, training and understanding, the problem is more deeply rooted. Designers seldom involve security in the process of designing which is another reason for so many vulnerabilities in the software. The most common security problems of today are Buffer Overflow and DoS attacks which were discovered nearly 50 years ago [Schenier 2008]. However, blaming designers for all security problems is just a part of the

picture; in reality, the current process of threat modeling is an iterative complex process. This dissuades a lot of designers from involving themselves in the process, since current processes are not intuitive or easy to master.

Problem Statement 3. Is there a possible means to provide designers with prescriptive methods to ease security analysis and threat modeling?

1.2.1.4 Where All to Apply

Threat modeling is moot if not applied at the correct level of abstraction for systems. While seemingly trivial, this is an important problem. Security does not have to be perfect, but risks have to be manageable. To manage risks one has to analyze threats at the correct levels and correct entry points. Avoiding threats is binary, either you are immune or vulnerable to threats, but the risks of some threats may be more than others. Of course, the most trivial solution would be to apply threat modeling at all layers, all avenues of a system. This would be pointless since as mentioned in Sec. 1.2.1.2, a defender needs a focussed view of the threats so as to make the right choices.

Problem Statement 3. What are the best entry points for threat modeling to be effective?

1.3 Summary of Contributions

The technical contributions made in this dissertation towards threat modeling of emerging networks are briefly outlined below:

Threat Framework: A threat framework that can be effectively used towards practical risk assessment is needed for formulating defense strategies. The work presented here provides a framework that enables a defender in assessing the risks to her system, by capturing and understanding the perspective of an attacker. The attacker's perspective is captured by analyzing various

aspects that need to be considered for a successful attack. The effectiveness of this approach is validated by studying various attacks and classifying them qualitatively as well quantitatively.

Identification of Attack Avenues: The threat framework presented in this work, can be utilized at any abstraction levels of a system. However, to effectively assess the threats, this dissertation identifies abstraction levels where it needs to be applied. The identification is done on the basis of the target characteristics, such as understanding which aspect of a system is under threat or at risk. This enables a more accurate and deeper understanding of the system and the risks faced by it. Further, some new avenues of attack are discovered such as the attractiveness of smartphones as attack devices, which forces a completely new approach towards defense strategies.

Attack Modeling: Various modeling techniques that are useful in the actual practice of threat modeling are provided in this dissertation. Modeling attacks in terms of their attack semantics, cost analysis and impact are some of the aspects that are studied. Some of the attacks that are modeled are also practically implemented and presented to provide better understanding of the interactions within various components of a system. One such work in this dissertation uses elements of epidemic theory to analyze the impact of malware propagation in social networks. The modeling of attacks provides insights on defense and mitigation techniques as well as understanding the building blocks towards proactive defense systems.

Classification of Attacks: Assessment of risks are useful only if they can be classified in some meaningful way; by using the framework and attack modeling this dissertation classifies various attacks in literature as well as some attacks conceptualised for emerging networks and applications. The classification of attacks is based on the impact, cost analysis as well as identification of the attack avenues.

Emerging Network Threat Modeling: In theory certain existing attacks can be easily modified for emerging networks. However, the defense strategies might not provide effective solutions, due to the various interactions of

system components or architecture. This dissertation uses the very aspects of attack modeling and the threat framework to provide threat modeling for emerging networks such as social networks and smartphones, which have not been modeled before, providing a start to defense techniques before the networks become popular.

Practical and Effective Techniques: Attacks and their solutions must be amenable to practical implementation. The work presented here is a step forward in this regard. The assumptions made towards attacks in prior research are abandoned, and the approaches are viewed from a real-world basis. The dissertation also showcases how some of the attacks cannot have practical solutions, unless the entire infrastructure is altered.

1.4 Roadmap

Succeeding chapters analyze the problem areas outlined previously in greater details. Prior to this, a survey of previous research efforts relevant to the problems are discussed in Chapter 2 along with research work at various levels of abstractions, based on some of the existing in the literature. Chapter 3 examines in detail the requirements of a practical threat modeling system and the necessity to incorporate the attacker's perspective in threat analysis. A generic framework is presented in this chapter based on the analysis of threat model requirements and attacker perspectives which also presents several attack avenues, thus providing the abstraction levels of a system where the threat model can be best used. Chapter 4 presents a threat model at the architecture level of abstraction, where the threats towards smartphones are modeled. A comparison of two popular flavors of smartphones is also presented and an analysis on the security characteristics provided. The threat modeling of smartphones also provides an insight to the capability of these devices as attack tools, which is presented in detail. The work in Chapter 5 presents the application of the threat model at the protocol level to assess the risk of a jamming attack at the MAC Layer. The chapter also models and analyzes the characteristics of a jamming attack and uses this information towards classifying the threat of such an attack towards wireless devices. Chapter 6

investigates and analyzes the impact of malaware propagation using the popular online social network Twitter. This chapter also presents several incremental attacks that take into consideration real-world usage and which can be used by a miscreant towards this goal. Attack models are created for each of the attacks to understand the mechanisms of such attacks, and a thorough cost analyses is used to calculate the feasibility of said attack. Chapter 7 discusses the work of this dissertation, the implications of the models, open-research issues and identifies avenues for future work.

2

Background and Related Work

2.1 Overview

In this chapter, a survey of prior research techniques pertinent to this dissertation is presented. The presentation is organised into sections, each focusing on a different aspect of the overall research problem. The first section highlights work done so far for better threat modeling and also provides details on the deficiencies in them. The second section motivates the problems in a network based on new architecture i.e., smartphones and presents some of the work conducted in this domain. The third section focuses on the attacks at a protocol level and provides the details of research work to mitigate this. Finally before summarizing this chapter, security research on the social network application is presented.

2.2 Threat Modeling

One of the fundamental problems in computer security is understanding at what levels of abstraction one can apply threat modeling to, in order to understand the threats that an attacker can exploit. This dissertation identifies three levels of abstraction used for threat modeling, viz., -

1. Architecture – This level presents the architecture of an underlying system or network such as VANETS [Raya and Hubaux 2005], smart grids [Massoud Amin and Wollenberg 2005] and smartphones.

2. Protocol – This abstraction level denotes the threats that network protocols inherently possess such as weakness towards jamming, partitioning, etc.

3. Application – At this level, the threat modeling is applied to the application as a whole. It has to be noted that in spite of having secure components, some applications may themselves present vulnerabilities.

A threat model, however, also needs to be generic so that it can be applied effectively at the above levels of abstraction. This dissertation first presents a generic threat model and the threat model methodology at the different levels, which have been a topic of research by other researchers. This chapter first reviews current approaches to threat modeling and presents some of the techniques that have been proposed. A review of the research at the different abstraction levels is also presented and the difference between the approaches taken in this dissertation to those in the literature is highlighted.

2.2.1 Traditional Threat Modeling

Threat modeling involves the identification of entry points via a formal method. Some other approaches involve defining a privilege boundary as well as threat visualizations. Entry point identification is a process of determining all possible access points to the system. Privilege boundary mapping is the assignment of access rights to system objects and threat visualization is a formal representation of threats using techniques like attack trees, security pattern description, and attack nets.

Risk analysis enables the separation of the critical or major threats from the minor ones [Barbeau et al. 2005]. In understanding the risks, knowledge of the real threats helps place the complex landscape of security mechanisms in context. The evaluation in [Barbeau et al. 2005] is conducted according to three criteria: likelihood, impact and risk. The likelihood criterion ranks the possibility that a threat materializes as an attack. The impact criterion ranks the consequences of an attack materializing a threat. The likelihood and impact criteria receive numerical values from one to three and for a given threat, the risk is defined as the product of the likelihood and impact. Depending on the

numerical values received the risk is classified as minor, major and critical. First, while the approach is relatively simple the likelihood of an attack is based from the system administrator's point of view and does not consider the absence of *a priori* knowledge of the system that an attacker is likely to have. Second, the evaluation requires the administrator to have expert knowledge of target systems or existing exploits [Geer and Harthorne 2002]. Further, most risk analyses do not consider network characteristics and their effects. The aforementioned reasons contribute to the inadequacy of such evaluation techniques to correctly analyze risks.

The authors of [Gupta and Winstead 2007, Jha et al. 2002] state that an attack graph can provide a methodology for documenting the risks of a system when it is designed. An attack graph is an abstraction that divulges the ways by which an attacker can leverage the vulnerability of a system to violate a security policy. However generation of the graph also requires analyzing the system's purpose and attacker goals which are seldom easy. They also describe how one can utilize the concept of attack graphs in assessing how a multistage attack occurs, where an attacker tries to utilize the intrusion into a system as launching point for other attacks, provided her intrusion is undetected. Historically, attack graphs have been manually created by red teams as cited in [Won and Kim 2006]. More recently, methods for automating this process have been developed with model checking [Ritchey and Ammann 2000], [Sheyner et al. 2002]. Unfortunately, model checking typically suffers from a lack of scalability due to the complexity and size of modern networks [Cheetancheri 1998], [Sheyner et al. 2002]. Consequently, expert systems providing a numerical likelihood or probability, for each have been explored to offer a more sustainable approach to understanding exploits. [Cheetancheri 1998], [Dawkins 2005].

NetSPA uses a predictive graph [Ingols et al. 2006] to generate scalable attack graphs. In addition, Mul-VAL uses Datalog as a modeling language to join preconditions and postconditions [Ou et al. 2005]. A commercial product called Skybox Secure also does attack modeling [Meiseles and Reshef 2005]. It uses a forward chaining algorithm, along with a network model, and a set of access control lists to determine what exposures an attacker can reach. Other research takes advantage of monotonicity to increase scalability [Ammann et al. 2002]. Monotonicity assumes that a pre-condition of an exposure is never invalidated by

the post-condition of another, allowing for the creation of significantly smaller attack graphs. However, incorporation of network characteristics in traditional risk analysis can prove beneficial and provide the system administrator with some information. The authors of [Duan et al. 2007] present a theoretical analysis of minimum cost blocking attacks on multi-path routing protocols in Wireless Mesh Networks (WMNs) and prove that such an attack is completely infeasible in WMNs. Their evaluation considers the effect of the attack, the characteristics of the target network such as traffic generation patterns and the size of the network on the attack. However, they too make certain assumptions such as the attacker having a way to implement the attack and *a priori* knowledge of the network. Traditional risk models and their assumptions illustrate the extent of the gap between the theoretical and practical risk analysis. This dissertation proposes the use of certatin parameters that affect an attacker in her attack to analyze the risks of attacks in order to bridge this gap. Attack surfaces [Manadhata and Wing 2010] provide an assessment of the degree of exposure of system components to untrusted parties. It is a conceptual tool used in increasing the security of a software during development. The authors of [Howard et al. 2005] provide details on how one can increase the security of a system by managing and minimizing the attack surface. However it must be noticed that in order to use the attack surfaces concept on a system, one has to know of all the possible vulnerabilities and then optimize the available resources to try to cover up the attack surface.

Broadly, threat modeling can be classified into two categories, namely, Attacker-Centric and Threat Centric [Mirembe and Muyeba 2008b]. In the following section, we highlight the difference in a broader sense before presenting indepth information of the models that make up these broad classes.

Attacker-Centric (AC): This threat modeling approach focuses on the identification of all possible access points to the system and the possible adversary aims. In general the attacker aim can be one or more of the following: Spoofing, Tampering, Repudiation, Information disclosure, Denial of services and Elevation of privileges (STRIDE) [Hernan et al. 2006]. STRIDE, as it is popularly known, captures only the intention of the adversary but not her capabilities and system defense attributes. By capabilities we mean, the potential tools, knowledge

and techniques the attacker might use to compromise a system. Most AC-based threat models are mainly visualized as attack trees [Schneier 2004] hence, they are simple to interpret and understand. Thus, AC-based threat models are popular among security experts although they lack adequate semantics to allow reasoning about threats they represent. Because of lack of adequate semantics, security controls developed based on AC approach suffice instead of being utility maximizing [Steffan and Schumacher 2002] (i.e., mitigate trivial threats but not the logical threats facing the system).

Threat-Centric (TC): A threat analyst employing Threat-Centric approach will focus on capturing system design and deployment flaws which can translate into security vulnerabilities. Threat centric approach provides a mechanism of examining system design principles and deployment configuration. In the TC approach, a threat analyst must step through the system design and deployment looking for vulnerabilities against each component of the design. TC threat modeling approach is the oldest technique of identifying vulnerabilities of a system and it has been extensively employed by mechanical engineers in the development of safety critical systems. Unlike AC-based models which have some semantics, most TC-based threat visualizations lack adequate semantics to allow reasoning about threats and their eventual validation. Thus for TC-based models to have a meaningful value, the threat analyst must synthesize sufficient background information about the system. In cases where sufficient background information is not available the effectiveness of the threat models drastically decreases. Most TC-based threat models are visualized as fault trees of the system. To be effective a threat modeling technique must capture both system attributes and attacker specific profiles while having sufficient semantics to enable logical reasoning about threats.

Threat Visualizations: Good threat visualization (representation) must capture both system specific attributes and time specific details. Therefore any threat model that is based only on either AC or TC is flawed because it is based on incomplete knowledge. An ideal threat visualization technique must be dynamic allowing the visualization of new threats as they appear hence making the security control adaptive. Also the visualization must be able to capture threats as generalizations often leave out critical information which results in flawed threat

models. In addition, the representation should be simple, easy to understand and interpret. Besides the simplicity, the visualizations should have sound semantics to facilitate logical reasoning about threats such as:

1. When are two threat paths equal?
2. What is the internal structure of threat?
3. Which particular event might have a greater impact even though it might have a low probability of occurrence?
4. What is the best way of synthesizing attacks?

For example attack suites [Won and Kim 2006] have sound semantics but are complex to understand and because of their complexity, they have not attracted much attention from security experts. In the following paragraphs some of the prominent threat visualizations against the afore mentioned desired attributes, is analyzed.

Fault trees are a graphical representation of interaction of system failures. The failures represent system vulnerabilities which present threats to the system. Fault trees were first published in the 1960's and have since then been employed by mechanical engineers in the analysis of system faults in mission critical systems. A node in the fault tree represents an event and the edges represent a causal-effect relationship between events. Leaf nodes are linked to the higher nodes in the hierarchy via logic gates (logic gate represent transformations). No-leaf nodes represent identified hazards for which predicted reliability or availability of data is required. Just like attack trees, intermediate nodes and leaf nodes represent refinements of a given fault. It has to be noted that fault trees lack adequate semantics to facilitate reasoning about threat models in addition to lack of expressiveness. Their lack of expressiveness is due to their inability to capture atomic details about the threat like attacker tools, knowledge, experience, motivation and goals. It is the limitations of fault trees as a threat visualization technique that has inspired the development of variants of tree-like threat visualizations structures like attack trees and attack nets.

The term *attack tree* was coined by Schneier [Schneier 2004] and it describes a directed graph which presents the why and how the security of a system can be compromised. In an attack tree every node represents an adversary goal and the root node represents the overall goal. Intermediate nodes in the graph represent sub-goals called (refinement of the parent goal) the adversary has to accomplish in order to achieve the main objective. Leaf nodes in the graph represent the atomicity of an attack i.e., sub-goals or goals that cannot be refined any further. Attack trees have simple semantics to allow the propagation of costs an adversary must incur to achieve a given task. However, semantics for attack trees have limited internal structure and can not facilitate sufficient logical reasoning about the threats they represent. In addition, the formalism must provide an avenue of incorporating system specific details while preserving the simplicity. The most pronounced advantage of attack tree is their simplicity of representation and hence interpretation.

In an attempt to improve the fundamental understanding of attack trees, Sjouke et al [Won and Kim 2006] propose an enhancement to attack trees by defining algebraic semantics. The researchers defined a universal set N of component whose various combinations can result into different attacks. Hence, an attack suite is a finite set of attacks. Their work introduced elegant semantics for attack trees which they transformed into attack suites. However their work did not address other concerns about attack trees like their static characteristics and being attack goal oriented. In addition, the semantics defined do not provide mechanism of synthesizing background knowledge in a logical way. Although the semantics introduced in attack suites are elegant, they introduced more complexity in the visualizations making the threat model difficult to understand. In general the complexity of the attack suites seem to overshadow the benefits of elegant semantics.

After analyzing the weakness in the ordinary hierarchical graphical approaches to threat representation like attack trees and fault trees, McDermott [McDermott 2001] shifted the threat modeling paradigm to Petri nets. The shift was inspired by the rich internal structure of Petri nets which offer more expressiveness by separating data and processes. McDermott defined an attack net as a Petri Net with a set P, where $P = p_1, p_2, ..., p_n$ of places and a set T, $T = t_1, t_2...., t_n$ of

transitions. Places represent state or known knowledge while transitions represent events or actions that might cause a change of state in one or more linked places. The places are linked to transitions by unidirectional arcs, which represent the cause-effect relationship. An attack net has a set of tokens S held in places and the movement of tokens between places along a given direction represent the progress of an attack. Attack nets present a departure from fault based analysis and attack tree threat representation by separating events from goals. The separation of events from goals enhanced the descriptive power of the representation, hence allowing security analyst to investigated atomic components of attacks. Despite the expressiveness of attack nets, the semantics of synthesizing information captured in the structure are not well defined. Furthermore, no comparison between attack nets and attack trees has been done to ascertain which one performs better than the other.

Security Pattern Descriptions (SPD's) are documents which describe the threat of a system in natural language. Security pattern descriptions are more expressive than graphical representations because they are not bound by formalism constrains. SPD's enable the capturing of atomic attributes of threats and background knowledge. Besides the expressiveness of SPD visualization, SPD represent threat models in a simple way hence, making the model easy to interpret. However, SPD visualization lacks adequate semantics to aid the systemization of threat models. Because of lack of semantics, SPDs are not ideal for automated tool builders, but they are very popular with security implementers. Therefore, there is need to define ways in which appropriate formal semantics can be incorporated into security patterns to enable logical reasoning about threats.

Another threat modeling technique is that of threat nets [Mirembe and Muyeba 2008a] that incorporates system design and deployment flaws, and attacker time specific attributes in the synthesis of threats. The model is built on foundations of Petri nets, because the inherent structure of Petri nets allows deposition of a node into three distinct components i.e., goal, background knowledge and events. A node (place) in the threat path is defined as a random variable, X, which represents a specific security service that might be compromised if a set of event(s) Y, Y representing transition (s)) below the node occurs. Background knowledge is quantified and represented as tokens in places. Since nodes are random variables,

the number of tokens per node in the tree hierarchy is nondeterministic. Arcs linking events to nodes reflect the progress of an attack in that direction.

A threat tree template [Morikawa and Yamaoka 2011] is essentially a redundant threat tree, fully loaded with a lot of potential attack scenarios. Concrete examples of attack techniques, common flaws,countermeasures, and their descriptions are included as well, to help users to understand the threat. The templates are handled in two phases: when a template designer prepares one, and when a template user (who is performing threat modeling) constructs a threat tree from it. It is assumed that template designers have appropriate security expertise, whereas template users are non-expert in security. In this section, the word "user" is used as a reference to a template user. A threat tree template is a tree composed of the following types of nodes: *Threat, Dependency, Example* and *Mitigation*. It is essentially a tree of Threats, optionally with nodes of the other types.

Threat. Threat node represents any threat. It must be a root node or a child of another Threat node. Every Threat node should be accompanied with a description written from an attacker's perspective. Sometimes it may be described as "an attacker can do something harmful by exploiting some weakness" if it is beneficial to mention a related weakness.

Dependency. Dependency means that its parent can be realized depending on another threat tree. Dependency must be a child of a Threat or Example node.

Example. Example represents an illustration of an attack or a vulnerability for the corresponding Threat. It must be a child of a Threat node or another Example node.

Mitigation. Mitigation optionally represents a common mitigation technique or countermeasure for its parent. It must be a child of a Threat or an Example node. Note that a child threat in the templates can be an elaboration of its parent, instead of only being a cause of the parent as in usual threat trees.

Risk assessment techniques often use threat source modeling. Both the NIST Risk Management Guide and the Morda framework utilize threat source models as cited by [Evans et al. 2004, Stoneburner et al. 2002]. In addition, the military uses threat source models for defense planning in both kinetic and cyber-warfare [Bell et al. 2005, Kewley and Lowry, Moore et al. 2001]. Others use a form of threat source modeling by assuming attackers will seek to acquire the greatest level of penetration on a host [Ammann et al. 2005]. Likewise, other attack graph research models present how a threat source will choose vulnerabilities based upon its propensity to avoid exploits that create IDS alerts [Sheyner and Wing 2004]. Another approach proposes the use of Bayesian networks to predict future attacker behavior based upon information gleaned from network sensors [Bell et al. 2005]. Other research identifies adversary profiling as a posteriori observable and threat source modeling as a priori [Lowry et al. 2011]. They also define and compare two classes of adversary profiling as named and class schemes, and propose using these models to drive attack graph creation. Although past research has identified the need for threat source models, there has been little guidance on how they can be created and used.

2.3 Architecture

As mentioned previously, existing attacks can be modified towards systems with a new underlying architecture quite easily. However, the same techniques cannot be used with defense or mitigation techniques as this could adversely affect the working of the system. This makes such systems attractive targets, since the technology is not mature enough and people are not well educated in their usage. One such system is the smartphone. The smartphone is becoming ubiquitous and is widely labeled to replace the conventional computing machines such as desktops and laptops.

Research on smartphones until recently, has been limited to investigating ways to utilize these devices for processing or using phone sensors for sensing applications [Beurer-Zuellig and Meckel 2008], [Lau and David 2010]. However, with the increase in smartphone's capabilities, the area of smartphone security has

also gained importance. Research has been made into studying the vulnerability of smartphones to rootkits and malware [Dixon and Mishra 2010], [Mulliner 2009] [Jin et al. 2008]. The authors of [Swami and Tschofenig 2006] describe how mobile phones can be intended targets of network attacks and provide a methodology for protecting these devices against TCP flooding attacks. However, their work considered the attacks over cellular network. Recently there also has been work relating to privacy and confidentiality [Loukas et al. 2010]. While all the research till now has looked at various aspects of security no one has really studied the effects of network attacks against the smartphones. Given the features and the popularity of current generation smartphones this dissertation investigates the effects of smartphones to different attacks. However, studying the effect of the attacks on smartphones is just one side of the coin. Given the fact that their capabilities could very well ensure that they become the primary device for most users, it is imperative to examine if these capabilities extend to being used as attractive attack tools. This dissertation presents a threat assessment of the attacks against smartphones as well as the attacks against conventional systems using smartphones which has not been presented before.

2.4 Protocol

Some of the most simple and effective attacks in the literature are against network protocols. The current generation of systems enlist various protocols and function based on their interactions with each other. This causes difficulties in assessing threats since the attacks exploit the inherent operation of these protocols. Thus, even creating effective defense or mitigation strategies becomes extremely difficult. One such attack is the jamming attack. Jamming attacks are a denial of service (DoS) attack that aim at disruption of either availability or freshness of data in wireless networks. Traditionally, denial of service attacks encompass either filling of user-domain or kernel-domain buffers Huang et al. [2003]. However, the wide availability of wireless networks and increased user-configurable wireless cards has led to users tweaking or changing the protocol(lower layers) of a device and hence control its behaviour. This has increased the threat of jamming attacks on wireless devices and various methodologies. Effects of Jamming attacks have

been extensively studied in Noubir and Lin [2003], Wood and Stankovic [2002], Wood et al. [2003], Xu et al. [2004]. The only effective solutions are in changing the MAC protocol or using expensive radio level technologies at the PHY level such as Direct-Sequence Spread Spectrum (DSSS) techniques. This dissertation tried to understand the working of this attack and assesses the risk these attacks present towards the modern generation wireless devices.

2.5 Application

One of the easier avenues of attack are a system application, since the weakest links in such cases are the human using them. The characteristics of such attacks are in tricking the human using these applications to install a "malware" which then exploits the vulnerabilities of an application. In this dissertation we focus on the malware propagation via Twitter an online social network. Malware propagation has been a long studied topic in network security. Malware propagation in scale free networks [Briesemeister et al. 2003, Griffin and Brooks 2006] has been investigated. Similarly propagation of malware in unconventional networks such as Wireless Sensor networks [De et al. 2009, Di Pietro and Verde 2011] and cellular networks (using MMS and Bluetooth) [Shin-Ming et al. 2011] has been studied. Malware propagation in traditional "social networks" such as email and instant messaging networks has been studied in [Mannan and van Oorschot 2005, Zou et al. 2004]. The worm and spammer based attacks on social networks have recently led researchers to focus on the security of online social networks. In [Faghani and Saidi 2009, Yan et al. 2011] the authors investigate malware propagation using simulated topologies and user activities. Similarly, the authors of [Xu et al. 2010] use correlation techniques based on user activities to suggest some mitigation schemes for worms in the online social networks context. A recent focus of researchers has been the understanding of how information flows in social networks such as Facebook and Twitter [Lerman and Ghosh 2010]. Authors of [Beck 2011, Benevenuto et al. 2010] have used this information to detect spammers in online social networks. Although, previous research has looked into some aspects of the security issues with online social networks, this dissertation presents a threat model and assessment customized to

Twitter. The use of game theory for social networks has also been studied before, albeit, briefly. Game theory is a powerful tool to study situations of conflict and cooperation, which is concerned with finding the best actions for individual decision makers (i.e., players) in these situations and recognizing stable outcomes. Games may generally be categorized as non-cooperative and cooperativegames. Non-cooperative game theory is concerned with the analysis of strategic choices and explicitly models the decision making process of a player out of his/her own interests. Unlike in non-cooperative games, in cooperative games, the players can make binding commitments. Game theory received special attention in 1994 with the award of the Nobel prize in economics to John Nash, John Harsanyi, and Reinhard Selten for their great contributions mainly in non-cooperative games. Game theory has been extensively used in modeling the effects of users who tend to deviate from the normal behavior in wireless networks [Key and McAuley 1999, MacKenzie and Wicker 2001; 2003] and especially in the widely used *IEEE 802.11* MAC protocol [Cagalj et al. 2005, Xiao et al. 2005, Zhao et al. 2007]. However, the agents in classical theory are assumed to be completely rational: they base their decisions solely on maximising their utility, are capable of performing very complex reasoning, all under the assumption of their adversaries being equally rational. However, human behavior is often times not rational, leading to scenarios where game theory cannot be used. In recent times, a new aspect of game theory has emerged called *Behavioral Game Theory* [Camerer], which examines how to effectively combine game theory and human behaviour. Unfortunately, due to logistical constraints and unpredictability of human behavior, the models have not yet matured. The first instance of modeling social networks using game theory was by the authors of [Kohli et al. 2012], where they tried to model how online social users would play a game of *Colonel Blotto*. The results suggested that under major simplifications,behavioral game theory can be an effective tool to understand social network behavior. However, the authors also clearly state that the model does not hold true, when the user cannot obtain knowledge of the strategy of the adversary. This suggests that behavioral game theory while capturing human behavior is still not able to model properly, if the opponent's strategy is not clear or visible. Similarly, authors of [Skyrms and Pemantle 2009,

Van Segbroeck et al. 2009] have also tried to model online social networks using game theory. A different approach over the years has been to try to use game theory in modeling security schemes relating to malware and its propagation [Khouzani et al. 2011, O'Donnell 2008, Singh and Lakhotia 2011]. The author of [?] uses a new model to estimate the timeline of malware for new operating systems. Similarly, other works [Shim et al. 2012] try to model the behavior of hackers and try to model effective incentives to prevent security researchers from becoming hackers or writing malware. It still remains to be seen if game theory and behavioral game theory in particular can be used to effectively design security schemes towards prevention of malware propagation. However, an incentivised approach might be feasible. This dissertation does not focus on the game theoretical modeling of malware propagation.

2.6 Summary

This chapter presented a survey of threat modeling techniques as well as the deficiencies in them. Surveys of research at the three levels of abstraction, viz., the architecture level, the protocol level and the application level, which also make up the most attractive attack avenues, were also presented along with the difficulties in assessing the risks as well as the motivation of why these problems are important in the current context of security systems. The following chapter presents details on a generic threat model that can be used as a framework for threat assessment.

3

A Generic Framework for Threat Assessment

3.1 Overview

As described in the introduction, threat modeling techniques suffer from various drawbacks. The biggest drawback in contemporary threat modeling approach is that threat modeling techniques are not generic or intuitive enough. Further, threat assessment usually occurs from the defender's perspective which is wide, since she wants to defend the entire system against attacks. In contrast, an attacker might now have such a perspective. An attacker's focus is on one particular vulnerability or a section of the system and tried to exploit it, thus making her perspective narrow and focused. As a consequence, defense techniques are not always feasible or appropriate to protect a system. This chapter presents a paradigm shift for the threat assessment approach in the sense that the dual perspective of both the defender as well as the attacker is taken. The rationale behind our approach is to take into account a key aspect of attack, viz., information and the cost of information required by an attacker to launch an attack. This crucial aspect appears to be largely oversimplified by researchers and leads to incomplete and ineffective threat assessment models. Further, the threat model presented in this chapter is generic, flexible, intuitive and attempts to consider some of the real-world issues that are necessary for accurate threat assessment of systems.

3.2 A Generic Framework

3.2.1 Assumptions of the Framework

The assumptions made by a model have a direct effect on the analysis of risks and can cause unreliable assessments. This can lead to a false sense of security or cause inefficient resource allocation by a system administrator. To clarify the statement let us look at the following example – Assume that an attacker could gain physical access to a target network or know beforehand the details of the network protocols running on the system. While this may seem as a valid assumption, it is evident that the risk assessment is highly affected when we include none, just one or both of the aforementioned assumptions. On the other hand if we were to assume that an attacker lacks the resources or technical knowhow on jamming a network, the entire risk assessment is a moot point no matter how good the model or the evaluation process is. Hence for an accurate risk assessment we need to objectively and fairly make assumptions without giving an attacker a clear advantage or disadvantage. For this reason while stating any assumptions it is important to keep them as close to a real world scenario as possible.

In this dissertation it is assumed that the attacker has no or very little *a priori* information about the target network. This includes knowledge about network components, its purpose or its usage. However, the attacker does have the resources and technical knowledge of implementing an attack and can gain the knowledge of the system he intends to attack. This is a valid assumption as discussed in Section 3.2.2. In this dissertation the same constraints are applied on the hardware the attacker possesses as in the real world. This however does not imply that the network is physically isolated, in the sense that an attacker is quite capable of both performing active and passive attacks on the target network. The scenarios of insider attacks and attacks resulting due to the mistake of a target network's user is not considered and is beyond the scope of this dissertation.

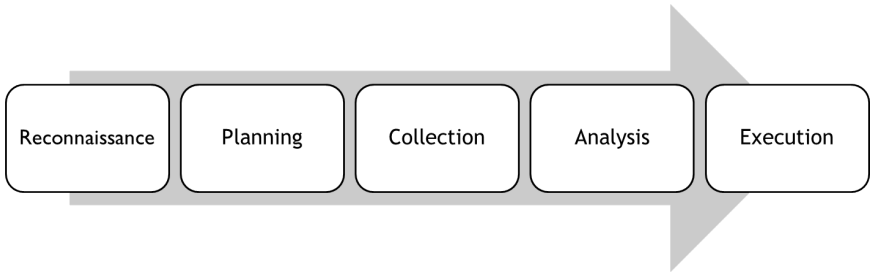

Figure 3.1: Steps involving an Attack

3.2.2 Modus Operandi of an Attack

Before the steps of an attack are presented, there is a need to clearly define an attack. An attack is a series of intentional steps taken to gain some unauthorized result. Since the steps of any attacker are intentional and methodical, it should be generally quantifiable and can be represented as a process, which in turn would help in creating a proactive defense strategy. Figure 3.1 presents the procedure followed by an attacker while targeting a system [Peikari and Fogie 2002]. The goal of these steps is to first obtain the Information Content necessary for the attack in order to execute an attack. Thus, the procedure to gain information about the network, is the precursor to an attack. From an attacker's point of view, this would include gaining as much information of the system as one can so as to develop one's strategy for attack. What we can broadly classify as information content are the features of the target network such as the data in the network, components, protocols of the target network, etc. It is important to understand that from an attacker's perspective this information content comprises of all the factors that has to be considered for staging an attack. Section 3.2.2.1 presents a detailed analysis and motivation of these factors. While the exact amount of information required for an attack depends on the skills of the attacker it can be fairly assumed that most of this information is essential for an attacker. In the current literature so far, it is usually assumed that the attacker already has the required information content. However, in this dissertation it is believed that if a defender has to regain her advantage this would be the starting point.

3.2.2.1 Motivational Factors of an Attack

The goal of any risk model is to assess the risk of an attack and classify the threat it poses to a network. However, from the defender's perspective the risk of an attack should relate closely to a real world scenario so as to be able to efficiently allocate her resources. In most cases the risk analysis of an attack takes into account only the defender's perspective and knowledge, and presents a rather pessimistic scenario. However, if the factors from the attacker's perspective is considered as well, the parameters that affect the analysis of a risk change. When both these perspectives are considered, the risk of an attack depends on:
i) Motivation of attacker, ii) Probability of attack, iii) Easier alternative, iv) Target network characteristics and v) Cost of attack. This is described next.

3.2.2.2 Motivation of an Attacker

This parameter directly affects the risk assessment of an attack and asymptotically either elevates or depreciates the risk of an attack. It is scientifically difficult to quantify this parameter as it depends on an attacker's behavior. However, one can try to quantify it by observing other factors such as the type, target and the purpose/effect of the attack. In [Barbeau et al. 2005], the authors state that an attacker's motivation can be categorized to be High, Medium and Low. Thus, both the purpose of attack and the motivation contribute to the overall risk of an attack. For example, a highly motivated attacker attacking out of inquisitiveness is likely to be less dangerous than one for financial gain.

3.2.2.3 Probability of Attack

This parameter denotes if an attack is desirable based on two factors – *cost of an attack* and the *severity factor* of the attack. The term *cost of attack* is defined as a combination of time, the hardware needed and the general strategy required for an attack. Severity factor is defined as the effect an attack has on a network. It is evident that the probability of an attack is likely to increase as the cost decreases and the severity increases. Thus the probability of attack can be quantified as

$$\Pr(Attack) = f(Severity_{Attack}, Cost_{Attack}) \tag{3.1}$$

3.2.2.4 Easier Alternative

This parameter relates the risk of an attack to another attack which is at a higher probability due to either increased severity or lower cost for a given network.

3.2.2.5 Target Network Characteristics

This parameter describes the features and characteristics of the target network. It encompasses other features such as system level misconfiguration [Sheyner et al. 2002], the unexpected side effect of operations [Chen et al. 2003] and platform specific attacks which can be exploited. Another factor that would be considered by an attacker is the type of traffic flowing through the network and the way it is generated.

3.2.2.6 Cost of Attack

This parameter quantifies what it would cost an attacker to launch an attack. The three factors that make up this parameter are *Time*, *Strategy* and *Hardware*. It is evident that the first two factors are directly dependent on each other and it is the prerogative of an attacker to decide which factor is more important to him. These two factors affect the third factor – as the attacker has to invest in the appropriate hardware depending on which of the above two factors he gives more importance to. Further, though we could assume that the attacker is not constrained by the type of hardware, it is possible that he is constrained by the hardware in terms of resources used and the performance of the hardware.

Time: This parameter denotes the time taken for an attack which includes the time for gathering information and implementation.

Hardware Constraints: This parameter specifies the constraints that an attacker has to both work with or face when launching an attack. The rationale behind this parameter can be motivated as follows – suppose an attacker takes over a node in a Wireless Sensor Network, the energy constraint as well as the

memory constraint would be a factor that would prevent him from making more complex attacks. On the other hand the same constraints (as the characteristic of the target network) also allow him in implementing a denial of service attack. Similarly the uncertainty of radio ranges [Zhou et al. 2004] and radio hardware could effect the severity of her attack.

Strategy: This parameter features in the cost of an attack and is an important parameter. It can be further subcategorised into:

Practical Difficulties. This factor considers the remaining aspect of difficulties while dealing with network hardware such as synchronization [Dolev et al. 2009] and basic cryptography in networks. This factor is also used to represent the unpredictable behavior of the wireless medium which equally affects the attacker as the target network such as radio ranges.

Implementation. This refers to implementation difficulties of attacks due to built-in defenses in the target network or hardware constraints.

Selection. This denotes the methodology of the attacker including factors such as gaining information content by gathering and storing data, analyzing it to obtain target network characteristics, and verifying the results. Too aggressive methods of gathering data, could unintentionally alert a system administrator about the attacker's intention. The information content includes operating system, hardware, type of data, network protocols, purpose of network, size of network, topology, etc. We are specifically interested in identifying a network protocol which contrary to intuition, is much more complex. For instance, the author of [Fall and Floyd 1996] suggests that the difference among NewReno and Reno (TCP) can be discovered only when multiple packets are dropped within the same congestion window. This suggests that the time and resources required by an attacker to accurately assess a network protocol are important. In the following section the tools and challenges in identifying network protocol are presented and the importance of this discussion is motivated.

Identification of Network Protocols. The correct functioning of a network protocol relies on specifications and implementations [Lee et al. 1997; 2002]. However implementations are inherently more complicated and could introduce discrepancies and vulnerabilities [Watson et al. 2004], even though the analysis for soundness validation may not discuss it [C.Meadows 1992, Lowe and Roscoe 1997, Meadows 2003]. It has been shown that most Internet protocols such as ICMP, TCP, TELNET, HTTP are subject to these discrepancies [Beverly 2004, Fall and Floyd 1996, Shah. 2004, Yarochkin. 1998]. The universal presence of these discrepancies is due to the fact that network protocols cannot be completely and deterministically specified; instead opportunities are provided for implementations to distinguish itself [Shu and Lee 2006]. The author of [Shu 2008] states that the identifying protocols employs the following two methods:

- Network Protocol Fingerprinting: This is the process of identifying a protocol by analyzing the output characteristics and traces based on the input given to the protocol using tools like NMAP [Yarochkin. 1998] or TBIT [Padhye and Floyd 2001, Shu and Lee 2006]. Here one can select an input with the aim of getting distinguished output traces. However this method called active fingerprinting is also prone to alerting system administrators. Passive fingerprinting, where one does not control the inputs but only observes the output traces is a time intensive process. Further, it is extremely difficult to conduct rigorous proof about the validity of fingerprinting experiments [Lee and Sabnani 1993]. The authors of [Shu and Lee 2006] show that the complexity and time required for fingerprinting makes it infeasible.

- Network Protocol Fuzz Testing: This is the process of mutating the normal traffic to reveal unwanted behavior such as crashing or confidentiality violation [Arkin and Yarochkin 2002]. However the authors also states that due to various factors this method is also mostly infeasible and inaccurate.

Figure 3.2 shows the risk model with the underlying factors and the relations which have been discussed above. This framework presents a threat model that is concurrent with our philosophy of threat assessment by taking into factors that an adversary has to consider for a successful attack and yet, is generic to be applied at

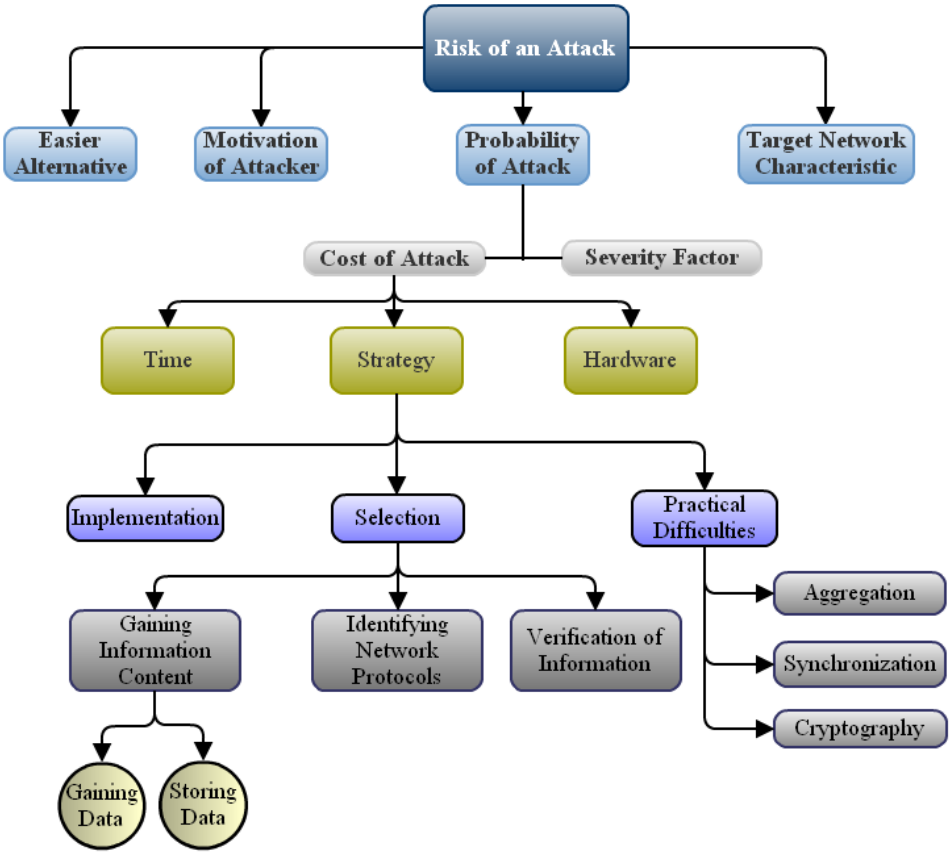

Figure 3.2: Relation between Factors affecting the Risk Assessment in our Model

different abstraction levels. The creation of such a framework is important since it also employs intuitive informal methodology that will encourage designers to be involved in threat assessment and can thus lead to the creation of stronger and more secure systems.

3.3 Summary

This chapter presented a generic framework for threat modeling that captured a dual perspective to the problem of threat modeling. Such a threat model will help designers in building stronger systems and also help system administrators in identifying and prioritizing attacks that they need to defend their system against. The threat model helps in reducing the disparity between attackers and defenders and help computer security research in general. While this dissertation examines threat modeling and its details in some emerging technologies in the cyber domain, it is important to present a generic framework to understand some of the parameters of attack. The details presented in this chapter and the generic framework presented makes it easier for a defender to fit various attack models, thus enabling an easier understanding and accurate assessment of the attack vectors. The following chapter presents the use of this framework as well as the dual perspective towards assessing risks at the architecture level, by examining the threats to smartphones.

4

Threat Modelling at the Architecture Level

4.1 Overview

Today mobile devices have become ubiquitous due to their ability to provide a wide variety of services to users. Current generation mobile hardware and operating systems, such as Apple iOS [Apple], Google Android [Developers, Enck et al. 2009], Windows Mobile 7 [Microsoft], and Blackberry RIM [Motion], have begun to rival personal computers in performance as well as capabilities. However as mobile devices and their operating systems continue to grow and evolve, security becomes a major concern due to the possibilities of these devices being threatened and exploited in manners that are similar to modern computers. The fact that these devices are so common makes them easy potential targets for an attacker and could be thought of as an attack platform similar to any other computer. Mobile devices often times store personal data, such as contact details, SMS messages, etc., as well as utilize a variety of applications that can prompt for and store additional information. This information can be valuable and potentially damaging if an attacker can retrieve it from the mobile device that it is stored on. Mobile devices would also make attractive aggressors since by default they are small in size, making them easy to conceal, highly portable, and are yet to be truly characterized as devices to launch attacks from. This dissertation looks at both the vulnerabilities of these devices as well as the attack capabilities of the

current generation smartphones through a Wi-Fi connection. First, details of the Secure Shell (SSH) vulnerability that exists on jailbroken iOS devices is motivated before presenting the details of an investigation of the existence of a similar exploit in the Android operating system. Second, an investigation regarding the possibility of using these devices in Denial of Service (DoS) attacks through SYN ACK and ping flooding is presented. Details of a tool that we developed, which is capable of finding devices on a wireless network and carrying out these attacks are presented. Finally, cases where computers can be used as attack devices to exploit vulnerabilities of these devices are examined and a comparison as to which *class* of devices would make better attack tools is presented.

The contributions of this chapter are as follows:

1. Investigating the SSH vulnerability that affects iOS devices.
2. Studying the feasibility and effects of DoS attacks against smartphones.
3. Investigating the scenario of smartphones as attack tools.

4.2 Exploiting SSH on Mobile Devices

One of the capabilities that users may add to a mobile device is the installation of a SSH server. This allows the user to remotely access their device without having to physically tether it to a computer. While this practice may greatly increase the ease of accessing files it requires that device be jailbroken. The jailbreaking or rooting process (depending on the device's operating system) allows for the installation of unofficial and/or unauthorized applications on the device. It is important to note that jailbreaking and rooting are frowned upon by both manufacturers and mobile network carriers, and typically voids the warranty.

4.2.1 iOS Jailbreaking and SSH

In order to gain the ability to remotely access an Apple iOS device through SSH, the device must first be jailbroken. Jailbreaking allows the operating system to gain root access and install applications that have not been approved by Apple's Certificate Authority. The process generally exploits some security flaw that is

present on the mobile device. Early generation iOS devices could be unlocked by exploiting the libtiff vulnerability, which is further detailed in [Pandya and Stamp 2010]. Current generation jailbreaking relies on exploiting vulnerabilities present within the bootrom which typically needs a hardware revision to patch. Usually the process of discovering and exploiting these vulnerabilities is generally difficult for the end users as such. However, the process has been made relatively simple for end users by groups which have developed some tools for the same. Some examples of popular jailbreaking tools are the iPhone Dev Team's redsn0w, PwnageTool [BLOG 2011], Greenpois0n by the Chronic Dev Team [DEV 2011] and Sn0wbreeze by ih8sn0w [ih8sn0w.com]. While jailbreaking an iDevice (devices that run on iOS) users are given the option to install Cydia [Freeman 2010] an open source alternative to Apple's App Store. Cydia's installation comes packaged with an SSH server, but users also have the option to install the OpenSSH application through Cydia.

4.2.2 iOS SSH Vulnerabilities

Jailbreaking an iOS device opens a potential vulnerability that can be exploited especially when an SSH server is actively running. Since the early versions of iOS, the superuser and password are assigned the default values "root" and "alpine" respectively, and with this knowledge anyone can connect to an SSH enabled iDevice as a root user. Using both the root access and file transfer capabilities of SSH, an attacker could steal any amount of information from the device, push files such as rootkits, malware, etc., on the device, or execute any system call. To exploit this, attackers need only scan a wireless network, find iOS devices running SSH and attempt to connect using the default username and password. This process is fairly simple using any network. While this vulnerability is fairly simple, it exists because of two main reasons: (1) the Jailbreaking process makes it very easy for even novices to install third party applications, and (2) many users who jailbreak their iOS devices often times do not know how to or care to change their superuser password or take other preventative measures that are available to them.

4.2.2.1 Remedies Against iOS SSH Vulnerabilities

There are several methods that users with jailbroken devices can use to prevent attackers from exploiting the above vulnerability. The first and most simple remedy is to change the root password, which can be done through the passwd utility or by directly modifying the master.passwd file. Of course this method is still subject to dictionary attacks if weak passwords are used. It is also possible to set up RSA key authentication over SSH instead of the use of passwords. This does involve additional steps such as the generation of the keys and their proper placement on devices (which may prevent some users from attempting this process). It would also be helpful to kill the SSH daemon when its utilities are not needed by the user. This would greatly minimize the risk from having the service constantly running.

4.2.3 Android Rooting and SSH

Unlike iDevices, the Android based devices do not need root access to install and use an SSH server. This is because developers of applications can publish them after digitally signing them due to the lack of a rigorous certification process. There are several applications available on the Android Market, such as *QuickSSHd* [TeslaCoil], that allows a user to run an SSH server without the need of rooting the device. While it is not necessary, one still can root an android device by exploiting the firmware and/or hardware. This makes the rooting process slightly more difficult because manufacturers often customize the Android operating system to suit their specific hardware platforms. Due to this all the "flavors" of Android may not have the same vulnerabilities. Also manufacturers have begun to release hardware with security features to prevent rooting, often times causing the device to fully fail if the process is attempted. One example of one such tool used to root several HTC mobile devices is *Unrevoked* [unrevoked]; it utilizes an exploit found by Sebastian Krahmer. Once the device is rooted a manual installation of an SSH server, such as Dropbear is possible. Unlike the iDevices, the SSH daemon will begin on restarting the device.

4.2.3.1 Android SSH Vulnerabilites

Smartphones that are running the Android operating system that is rooted and have an SSH server, do not have the same vulnerability as iOS devices. This is due mainly because of the fact that either no superuser password exists or can be found. Hence, there is no default set of values for an attacker to use. Furthermore, in the case that the user manually installs Dropbear, he has to specify the superuser name and password values. For the more advanced users, Dropbear can be configured to use RSA key authentication making the vulnerability much more difficult to exploit.

4.3 Smartphones as Aggressors/Victims

While current generation smartphones have continued to grow more powerful in terms of both hardware capabilities and processing power than their predecessors they still have one major constraint, namely power consumption. As a result it may be attractive for an attacker to simply kill off the device, using, say, a denial of sleep attack, rather than attempting to steal information from it. By utilizing network attacks, an attacker can consume the battery life of the device by causing the radio to be constantly receiving and possibly transmitting. This chapter focuses on the flooding attacks, SYN and ping, in order to generate a great deal of network traffic focused solely on the intended mobile device.

4.3.1 Attack Overviews

SYN Attack: Unlike the more popularly known SYN flood attacks [Nakashima and Oshima, Nashat et al. 2008], a variant of it is used to generate a large amount of network traffic in a short time frame. In this variant of attack, one opens and closes TCP connections to the target device repeatedly and as quickly as possible. This will, instead of impairing the TCP service, generate a series of SYN, ACK packets in accordance with the three way handshake required at the start of every TCP connection. By immediately closing the connection once the handshake has finished, one can then restart the handshake by opening another connection. The underlying assumption while carrying out this method to generate traffic is that

the mobile device has some TCP ports open for connections. Without open TCP ports there would be no active services running on the device to connect to and hence no network traffic would be generated. This kind of an attack is implemented to target mobile devices primarily to leverage the power constrained smartphones. However, such an attack could also be easily carried out against other devices such as laptops.

PING Flooding Attack: Ping flooding [Cabrera et al. 2001] is an attack based on a built-in feature namely the ping command on most operating systems. However, only the Unix operating system supports the "-f" flag. When used with this flag, the ping command will send packets as fast as can be supported by the network or up to one hundred times per second, whichever the system deems is greater. The ping command sends an ICMP "ECHO_REQUEST" packet and waits for an ICMP ECHO_RESPONSE to be received from the host. By executing this command against a device one can generate a massive amount of network traffic targeting only the device. Ping flooding does not have the same downfall as the above attack since it does not require any open TCP ports to generate traffic.

4.3.2 Synthesizing Attack Scenarios

Two different scenarios are considered, the first being more powerful devices, such as laptops that can have a constant power (if required) attacking smartphones. This scenario conforms with a typical model, as we often expect the more powerful devices to act as an aggressor. As mentioned above, in this scenario we are concerned with depleting the smartphone's power since as outlined above, stealing information from the smartphone requires the device to be jailbroken, a situation we believe might not be very common.

The second scenario we investigate is the smartphone acting as the aggressor and attacking its more powerful counterpart such as a laptop. While traditionally one would not have the same expectations from this model as we do with the first, it is important to note that smartphones do have some advantages. First, as they

are not perceived to act as aggressors, they are likely to be less sought after when an attack is being carried out. This means they are more attractive to attackers as they can be concealed by an attacker. Further, since smartphones are resource constrained, many of their capabilities are optimized to be more efficient (in terms of memory and power consumption) than their powerful counterparts. However, the lack of additional features in terms of both hardware (such as USB ports) and software (firmware) also limits these devices. For example, smartphones cannot be used to sniff packets over the wireless medium unlike a laptop where an attacker can add network cards (via a USB dongle) and sniff packets in promiscuous mode.

4.4 Experimental Setup and System Model

All experiments were conducted in a controlled environment with only our devices connected to the network. This was done primarily to check the effects of the attack as well as the security policies of the university where this research was conducted. The smartphone used for our experiments was HTC's Droid incredible [HTC 2011]. The Droid Incredible by HTC comes with Android 2.2 and is equipped with a 1 GHz Qualcomm Snapdragon processor. The smartphone also comes with 512Mb RAM and is Wi-Fi IEEE 802.11 b/g capable. Two laptop computers, one with the Ubuntu OS and one with the Windows OS were used as the powerful counterparts to the smartphones. The Laptop with the Windows OS was used for the SYN ACK flood attack and ran a tool that is created for this purpose. The Ubuntu laptop was used for the ping flood attack (since the Unix system supported the "-f" flag we needed). The Ubuntu laptop was also the target for the attacks initiated from the smartphone. For the ping flood attack, the Windows system was used to observe the network speeds as as some possible side effects of the ping flood are envisioned. Before proceeding to the results of our experiments, the details of the tool created to automate the SYN ACK flood attack and also the attack procedures are presented in the following section.

4.4.1 SYN Attack Tool and Attack Procedures

In order to automate the process needed for discovering and exploiting mobile devices connected to a wireless access point a tool was developed. The tool was implemented in Java and provides the user a relatively straightforward interface to carry out the exploits. The backend of the tool relies heavily on the popular network exploration tool Nmap [NMAP 2010]. Using the tool one can perform an initial host discovery on a wireless network and determine all "online" hosts in a user specified subnet. It is important to note that this requires the device on which the tool will be used to be connected to the same wireless network. One can then use the command to perform the initial host discovery scan. An example of the command would be as follows: `nmap sP 192.168.1.100/24 exclude 192.168.1.100`. The details of the commands are as follows: the `sP` flag specifies to the tool to use Ping Scan, a common utility in Nmap. The next argument namely, `192.168.1.100/24`, specifies the subnet of the scan which by default is 24 but can be specified by the user. Thus, if the IP address `192.168.1.100` is that of the computer the tool is currently running on, the ping scan would check the IP addresses from `192.168.1.100` to `192.168.1.256`. The `exclude` is added to eliminate the host computer from the scan. This preliminary scan allows us to get the IP address, MAC address and NIC vendor (which is derived from the MAC address) for each host discovered on the network. Once a list of active hosts is obtained, a more intense scan can then be performed to find the open TCP ports as well as the device's operating system. The following Nmap command, `nmap O p1-65535 IP address` performs the OS detection scan on IP address and scans all TCP ports. This additional information is crucial when we are deciding which targets to attack. Once any open TCP ports on the host are discovered, one can start the SYN ACK attack and flood the device. Thus, the steps for performing a SYN ACK flood against a smartphone are as follows:

1. Start the tool. A host discovery of the default subnet is performed based on the user's IP address. The hosts that are found are displayed in a list format on the left side of the tool. The user can also select a new subnet and perform the host discovery again at any time.

2. Scan a host. After selecting a host in the list and pressing the Scan button, an OS detection and scan of all ports takes place and the results are returned to the user in the right window.

3. Flood a host. Once a host has been scanned, if any TCP ports are in an open state the tool can begin to SYN ACK flood by selecting the Flood button, however while the attack is running the tool cannot perform any other operations. The attack can be canceled at any time. It is important to note that the SYN ACK flood attack is an attractive option for an attack, since most smartphones lack a firewall and the effects of the attack are undetectable from the user's point of view. However as explained before it requires the phone to have some open ports.

The ping flood is not implemented on the tool since "ping" is an in-built utility present on Linux and Unix. To carry out a ping flood attack one can simply open a command shell on a Linux machine and enter the following command: `ping f IP address`, where `IP address` is the host that we wish to flood. The tool described above can be used to find the IP address of the intended victim. This technique of course does not depend on open ports and could be carried out after the preliminary host discovery scan.

4.5 Experiments and Results

In this section experimental results of the two scenarios described in Sec. 4.3.2 are presented. The first are the results of the attack on a smartphone using a laptop and next the contrasting scenario, where a laptop is attacked by a smartphone.

4.5.1 Scenario 1 - Laptop as Aggressor

In this section the methodologies and specifics of the attacks are presented before the results of the attacks.

Attack Methodology: As our attacks aimed at depleting the power of the smartphones, the experiments in two scenarios were performed. First to establish a baseline for the battery life of the smartphone needed to be determined in the absence of attacks. Next by performing the attacks on the smartphone the effectiveness of the attacks was determined. During all of these tests the device was turned on, started at full battery life, the screen was kept on at full brightness and the volume was kept at 50%.

While determining the baseline for battery life we needed to understand the power consumption of individual components and various network interfaces, to model the power consumption of smartphones in a typical day-to-day real-world scenario. Thus, the baseline battery life with only the 3G interface of the smartphone on was determined. Next, only the Wi-Fi interface was switched on but no network traffic was generated. Finally web traffic by accessing videos on YouTube using the Wi-Fi interface was generated in order to simulate a more realistic scenario. To determine the effectiveness of attacks by themselves the SYN ACK and ping flooding on the targeted smartphone without any additional traffic was conducted. Finally the SYN ACK flooding and ping flooding were conducted while generating web traffic to determine how effective these attacks are on a smartphone.

Results: Fig. 4.1 shows the smartphone's battery consumption under each of the scenarios discussed above. It is clear that both of the attacks are highly effective in draining the battery, however, the ping attack is especially more effective and attractive to an attacker. This is due to the fact that the ping flood requirement is less strict, unlike the SYN ACK flood that requires TCP ports to be open. Further, the ping flood also had a side effect on the network as it DoS-ed the network and network speeds deteriorated drastically. For thoroughness and to further study the effect of ping packet size (which can be user defined) the ping flood attack was performed for 20 minutes varying the ping packet sizes.

Fig. 4.2 illustrates the results of the above experiments. As can be seen, the percentage of battery consumption is greatest when the ping size packets are 1028 bytes and 2028 bytes. A further increase in the ping packet size does not result in an increase in the battery power consumption.

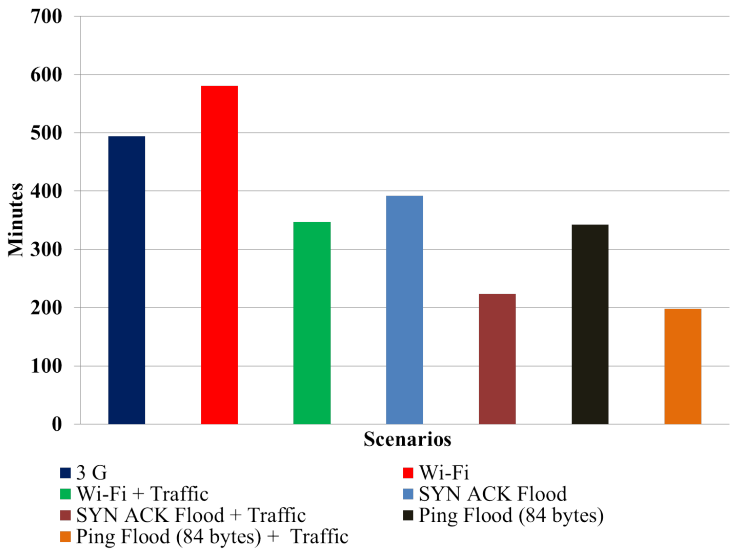

Figure 4.1: Smartphone battery consumption under various scenarios

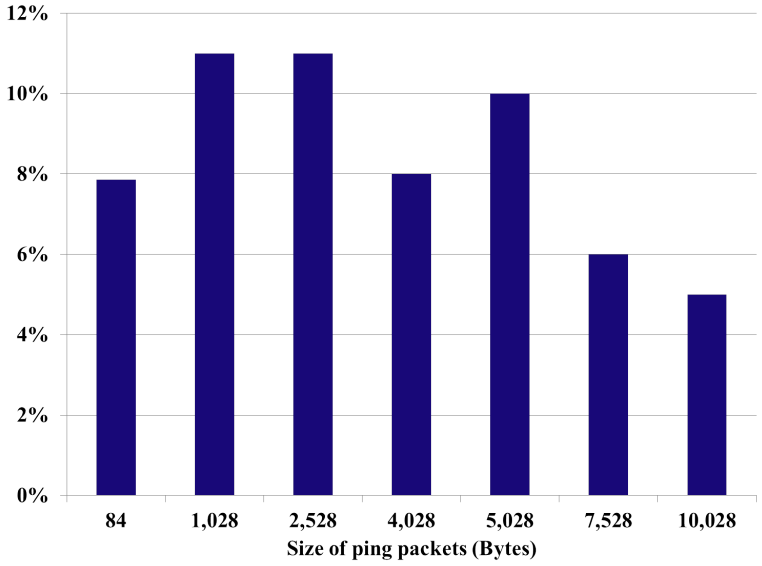

Figure 4.2: Effect of the size of Ping Packets on battery life when ping flooded for 20 minutes

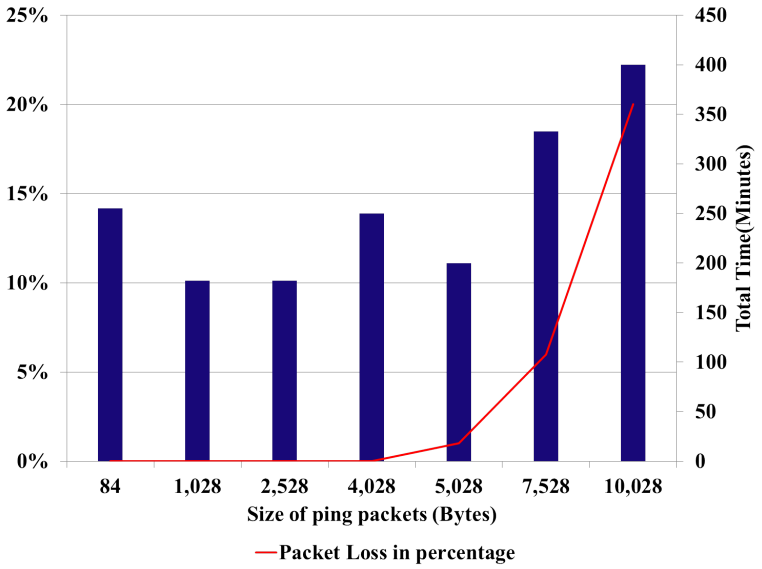

Figure 4.3: Battery Drain Time for different ping packet sizes and corresponding packet loss percentages observed

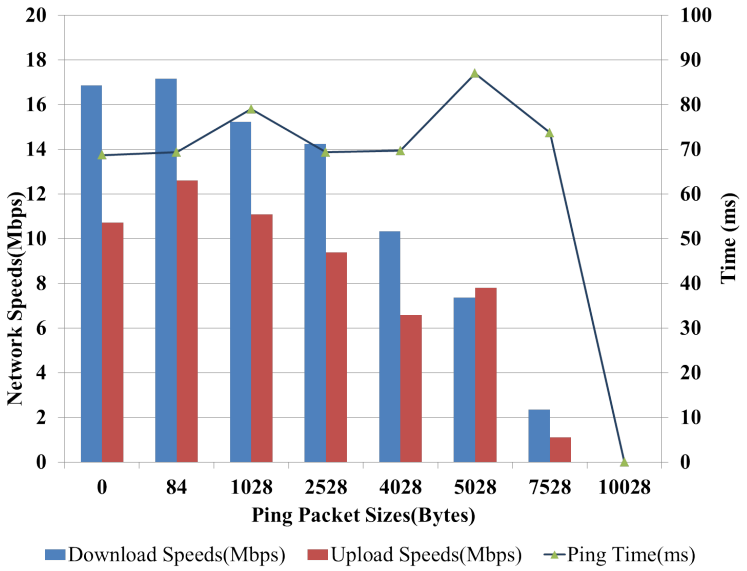

Figure 4.4: Effect of ping packet on Network Characteristics

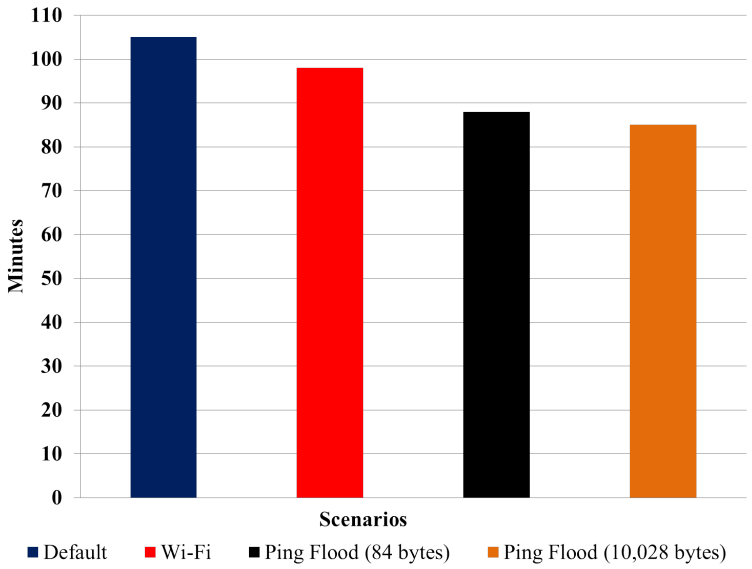

Figure 4.5: Laptop Consumption under Various Scenarios

Fig. 4.3 illustrates the results of our experiments and corroborates our observations. Fig. 4.4 depicts the severity of ping flood on the network as the size of ping packets increases. While these experiments show that the minimum time taken to completely drain the battery of a smartphone is a minimum of 3 hours, a ping flood attack would deny service to all clients connected to the network very quickly.

4.5.2 Scenario 2 - Smartphone as Aggressor

As before, in the following section first the specifics and methodologies of attacks are presented before the results of the attacks.

Attack Methodology: Based on our observations from the above experiments it was decided to perform only the ping flood attack. The attack can be performed as similar to the above case since one can access the command line prompt in Android with several applications.

Results: Fig. 4.5 illustrates the results of the ping flood attack on the laptop using a smartphone. It is interesting to note that the ping flood is not as effective

on the laptop as it was on the smartphone, since unlike in smartphones where the interfaces are designed to transition into "sleep" mode when not used, the laptop interfaces are always on. The DoS effect on the network remains the same.

4.6 Summary

In this chapter the security of smartphones and the SSH vulnerability of iDevices was investigated. This chapter demonstrated how Android devices do not suffer from the same vulnerability. Also presented were two novel attack scenarios involving smartphones and it was demonstrated that while smartphones do present an attractive target to launch DoS attacks against networks targeting the power consumption of smartphones is infeasible. Similarly a key finding in this chapter was that while the simplicity of the SYN ACK attack might make it an attractive choice, added with the lack of firewalls on smartphones, it requires too many conditions to be a successful attack and is thus infeasible. However, given the battery capacity and form factor of a smartphone, they make an attractive option as an attack tool. While the attacks presented show the capabilities of smartphones as both targets and aggressors, it is clear that with the growing smartphone segment, threat modeling towards and against these devices is very much necessary. The following chapter presents the details and risk analysis on jamming attacks at the MAC layer, which is one of the most simple attacks in the literature against network protocols.

5

Protocol Level Threat Modeling

5.1 Overview

One of the most important problems in the field of computer security is the jamming attack at the MAC layer for wireless networks. This problem has been studied by many researchers and while some solutions have been proposed they are not completely feasible. In this chapter, the generic framework introduced in Chapter 3 is applied to assess the threat of MAC layer jamming in wireless networks. The chapter also provides some insights into the amount of required information and cost of this information for an attacker to launch a successful jamming attack. The objective of the threat model of this chapter is to provide a basis for proactive security schemes in wireless networks.

5.2 MAC Layer Jamming in WSN

Jamming attacks are a type of denial of service (DoS) attack that aim at disrupting either the availability or freshness of data in wireless networks. Traditionally, DoS attacks encompassed either filling of user-domain or kernel-domain buffers [Huang et al. 2003]. However, the wide availability of wireless networks and increased user-configurable wireless cards has led to users tweaking or changing the protocol (lower layers) of the device to control its behavior. This has increased the threat of jamming attacks on wireless devices which have been extensively

studied [Noubir and Lin 2003, Wood and Stankovic 2002, Wood et al. 2003, Xu et al. 2004].

5.2.1 Preliminiaries

In the following subsection, the preliminaries of a jamming attack such as the characterization of what constitutes a jamming attack are presented before proceeding to the different profiles/mechanisms of jamming attack.

5.2.1.1 Characterization of a Jamming Attack

Jamming attacks target the Medium Access Control Layer (MAC) or the Physical (PHY) Layer of the OSI stack. This attack involves a jammer causing interference by emitting a RF signal continuously, disrupting the operations of a target network. However, the authors of [Chen et al. 2008, Xu et al. 2006] state that a broader range of behaviors can be adopted by a jammer and a common characteristic of jamming attacks is that their communications are not compliant with the MAC protocols. They define a jammer as any entity interfering with the transmission or reception of wireless communications by either preventing a source from sending out a packet or reception of legitimate packets, leveraging mostly on the shortcomings of the MAC or PHY protocols. Any attack based on this idea is classified as a jamming attack. Denial of Sleep [Brownfield et al. 2005, Raymond et al. 2006, Raymond and Midkiff 2008] and RTS/CTS jamming [Chan et al. 2007, Wood and Stankovic 2002] are some flavors of jamming attacks targeting WSNs, and are aimed primarily at depleting energy while data staleness is also an attractive byproduct of such attacks.

5.2.1.2 Profiles of a Jammer

The success of a jamming attack like most attacks is dependent on the strategy chosen by the jammer. It must be noted that the strategy in this kind of attack includes both the layer of choice, i.e., either PHY or MAC and the model used to *jam* it. There are four different models or profiles of jammers [Chen et al. 2008, Xu et al. 2006].

Reactive: A reactive jammer aims at disrupting the *reception* of a packet by transmitting a radio signal on detecting activity on the channel. The crux of the strategy is to cause packet collisions forcing the sender to retransmit. This profile requires continuous listening, thus consuming the jammer's energy as well. The authors of [Chen et al. 2008] state that these jamming attacks are difficult to detect and alleviate.

Constant: A constant jammer aims at continuously jamming a network by sending random bits over the channel forcing other nodes to either back off or constantly listen to the channel. An important caveat in this is that the jammer follows its *own* noncompliance to the protocol to prevent legitimate conversations and does not wait for an idle channel.

Random: A random jammer alternates between jamming and sleeping to conserve energy. It's effectiveness depends on the frequency or distribution of jamming and sleeping times t_j and t_s respectively. It is important to note that during the jamming phase the jammer can follow one of the remaining profiles.

Deceptive: A deceptive jammer, injects/transmits regular packets in between transmission of legitimate packets by other nodes and not random bits. This forces the other nodes to assume that a legitimate packet is being transmitted and they continue to be in the receive state.

5.2.1.3 Severity of Jamming Attack

Jamming attacks at the MAC level are effective due to the simple strategy and the difficulties in detection [Wood and Stankovic 2002, Xu et al. 2006]. Further since these attacks specifically target the protocols there are no effective means of circumventing the problem. Particularly, the problem lies in the inability of the network devices to distinguish between *malicious* jamming and *unintentional* interference. The only effective solutions are changes to the MAC protocol or using expensive radio level technologies at the PHY level such as Direct-Sequence Spread Spectrum (DSSS) techniques [Poisel 2006].

5.2.2 Effectiveness of Jamming Attack

From a network perspective the effectiveness of jamming attacks is dependent on the following two necessary features of the network.

Target Network Characteristics: WSNs or Ad-Hoc Networks are attractive targets due to their resource constrained nature since jamming attacks aim at depleting the energy of the devices by reducing their sleep times, increasing either the number or time of re-transmissions. Another characteristic of jamming is that it directly affects the data flow in a network making it effective against networks where data freshness is critical.

Hiding in Plain Sight: The success and effectiveness of the attack also depends on the jammer's ability to remain unidentifiable in the network. While a part lies in the implementation of the attack, a major part is the network's inability to differentiate between jamming and congestion. In addition to this it is also necessary that the network cannot identify the misbehaving devices. This implies that any kind of scheduled access to the medium is ruled out, as in such cases the jammer(s) can be easily identified and the network can differentiate if it is under attack.

5.2.3 Consideration of Jammer's Perspective

As explained in Section 5.2.1.3, the effectiveness and strategy of a jamming attack makes it hard for a network administrator to defend without investing in expensive countermeasures. Further, the countermeasures require an elaborate protocol of secret sharing for the scheme to be viable and effective. Considering this one would have to assume that such attacks would be nearly impossible to prevent or protect. However, the lack of evidence of such attacks in real-world [Peters 2009] implies that while theoretically plausible there are some caveats in this kind of attack that make them less popular with *attackers*.

A reasonable explanation as to why such an attack is unattractive to an attacker could be that the effort required for successful initiation of the attack requires large effort with diminishing returns or that the attack does not comply

with the motivations of most attackers. The following subsection analyzes which of the two factors is the reason for the unpopularity of such attacks.

5.2.4 Attacker's Perspective and Concerns

To understand why these attacks are unpopular it is imperative to understand the perspective of the attacker. Further, there are a lot of practical concerns that also need to be considered especially when the attacker starts off with zero knowledge about the target network. As explained in Section 3.2.2 some of these concerns fall under one of the many steps an attacker takes to increase the chance of success of attack.

To begin with an attacker has to spend considerable resources to ascertain that the network complies to the two necessary conditions described in Section 5.2.2. This includes finding the answers to the following questions:

1. What is the type of network? This critical question has to be addressed for the attacker to know what target network he is attacking.

2. Is the concern of the network energy or data freshness? This question would tell an attacker if a jamming attack is going to be effective or not.

3. What is the type of data flow in the network – Periodic, Query based or Event Driven?

4. If the concern is data freshness, what are the standard packet sizes that flow in the network? Are there other features in the network such as aggregation or network coding?

   Answers to the above questions help in choosing the kind of profile. Methods such as aggregation/network coding will reduce the effectiveness of the attack or require deploying/taking over more resources.

5. Identifying the *exact* protocol of the network. This is another critical dependency for an attacker. A motivating example for this is that the implementation of the attack is completely different in case of a CSMA MAC protocol from a preamble based MAC protocol. If the target network is running a schedule based MAC protocol, the attack will be ineffective.

6. Identifying physical access to the channel. What is the power required to jam? For example, if the devices transmit using BPSK or AM [Proakis and Manolakis 2006], due to the robustness of the signals the jamming attack may not be viable.

These are some of the concerns an attacker has to address to guarantee success to even an extent. However the following are also some additional practical concerns which an attacker needs to address.

1. What is the size of the network? What is its topology?

2. How to implement her attack? Does an attacker have physical access to the network? Where to place the jamming nodes?

Based on the analysis of a multitude of jamming attacks, the following steps are derived for describing the preparation of a jamming attacker as shown in Figure 5.1. The figure shows that there are 3 main steps for an attacker, namely, *Identifying Network Characteristics, Identifying Exact Protocols and Implementation Concerns.*

Section 3.2.2.6 describes the concerns and analysis of identifying network characteristics and exact protocols. We now focus on the implementation concerns for the practical aspects of the attack. The implementation of the attack requires us to consider two scenarios as shown in Figure 5.1 – Takeover Target Devices or Deploy Own Devices. We present an analysis below:

1. Takeover Target Devices: In this scenario, the attacker has to take over the nodes of the target devices and use them in her attack. Since we do not consider human interaction, an attacker has to get within transmission range or have physical access to the devices. In cases of WSN or Ad-Hoc networks tamper proof/resistant (TPD) devices [Ning and Du 2007] could easily circumvent this problem. Further, if physical access is possible, then the attacker has easier options such as destroying them which is a feasible alternative since it is a DoS attack.

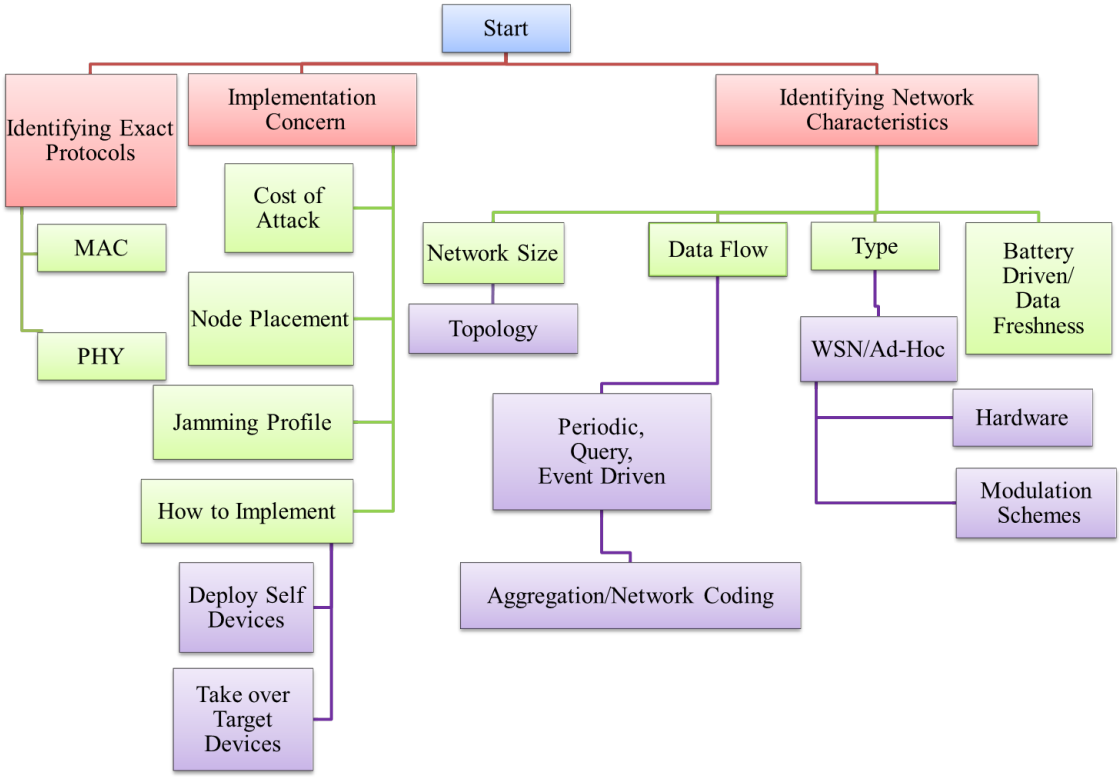

Figure 5.1: Steps an attacker has to take for a Jamming Attack

2. Deploy Own Devices: In this scenario, the attacker deploys her own devices. While this scenario is feasible and is likely to improve the success rate, the cost of attack increases. The attacker has to invest in the devices just for denying service or interfering in the performance. Again easier alternatives such as destroying devices exists. The scenario of a more powerful device (such as a laptop) against sensors does exist, however the effect of jamming would be localized to a small region. Further, even in such cases the attacker too is restricted with the same energy constraints. Deploying more than one laptop again is going to increase her cost of attack manifold.

The next important aspect of implementation is choosing an optimum jammer profile since all the profiles are orthogonal to each other in terms of effect.

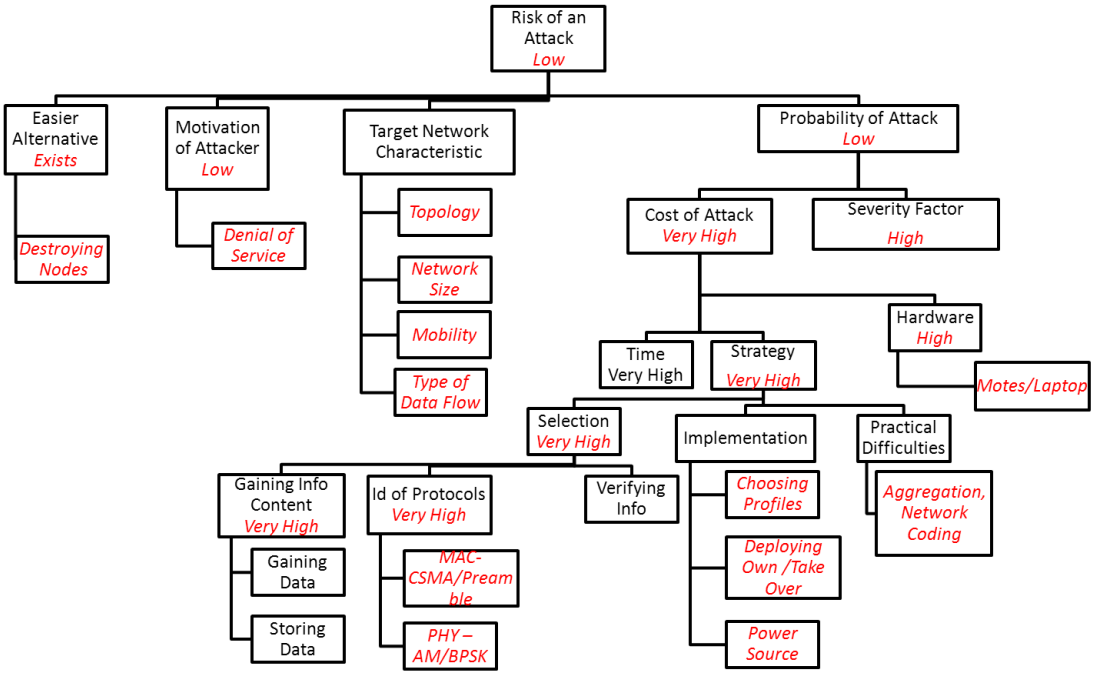

Figure 5.2: Risk Model Applied to Jamming Attacks

1. Constant: This profile is effective on all kinds of protocols. However, the type of data flow also directly affects its efficiency. If the data flow is periodic, event driven or query based, constant jamming is going to be wasteful and will also affect the life of the jammer nodes as they would have to transmit all the time.

2. Deceptive: This profile is very effective on a very small subset of preamble based protocol. However, this profile requires the jammer to be able to exactly ascertain the protocol as it has to send the exact preamble or the packet.

3. Random: This profile is the most efficient profile, provided that the jammer is able to configure the exact time/distribution of sleeping and jamming.

Its efficiency reduces significantly in Event Driven networks and would not be effective at all in query based networks. It is again important to note that this profile attacks data freshness more than energy consumption.

4. Reactive: This profile is the most effective but also the least efficient since the jammer node has to be "ON" all the time. While it circumvents the amount of information content required by an attacker, networks with aggregation or small packet sizes would not be really affected. Further, considering that the energy consumption for receiving is nearly equal to transmission, this profile would lead to unnecessary wastage of energy. Networks where there is a constant (or near constant) flow of data and data freshness/delay sensitivity is critical are most vulnerable to this profile.

The most important factor in this attack after observing the steps of a jamming attack is the cost of attack. This attack aims at a small subset of networks and requires too many necessary conditions for the attack to be successful. Simply put, this kind of attack extracts a huge cost in terms of time and resources from the attacker, due to the amount of reconnaissance required. As has been illustrated above, the attacker has to invest a lot in reconnaissance since even small mistakes could completely nullify her attack vector. For example, mobility of target devices or base station, would render this attack completely useless. Further, this kind of attack has very little return for the attacker for the amount of investments he has to put in.

5.3 Jamming Attack Risk Model

The description above leads to a risk model for jamming attacks as shown by Figure 5.2. This is an instance of the generic model from Figure 3.2 where the boxes represent the factors we have identified, with their respective values shown in italics.

5.4 Summary

In this chapter the generic framework introduced in Chapter 3 was applied to MAC layer protocols and the risk of jamming attacks was assessed. The risk assessment was *low* due to the fact that the attacker has better options and that for a successful jamming attack, the cost of gaining information is very high. Our findings are validated as the incidents of jamming attacks in the 2010 crime and security survey conducted by the FBI [Peters 2011]. The following chapter presents the threat model at the application layer, by analyzing the impact and the techniques of how Twitter, a popular social networking application, can be used for malware propagation.

6

Application Level Threat Modeling

6.1 Overview

Malware and its propagation is a difficult problem to solve. In the past, spammers used traditional "social-networks" such as emails and newsgroups enticing unsuspecting users to install and then propagate worms. The advent of Pay-Per-Install (PPI), which help "miscreants to outsource the global dissemination of their malware" Caballero et al. [2011] has led to a diversification of malware propagation attempts. One such target of these attempts is the on-line social networks such as Facebook and Twitter. Online social networks are Internet based schemes that could provide an ideal avenue for malware propagation since there are clearly defined paths already set up. Facebook has showcased its vulnerability when it was targeted by *koobface* Shin-Ming et al. [2011] and *clickjacking* worms Mannan and van Oorschot [2005]. Twitter has also been targeted in the past, but mainly by spammers who targeted overloading Twitter's servers Mannan and van Oorschot [2005]. The rise of such attacks, targeting online social networks leverages the facts that these technologies have not fully matured and its users are not completely educated on the risks. However, the risk of exploiting these technologies by the groups which promote either Pay-Per-Install or Pay-per-Click (PPC) and forcing unwitting victims into downloading and propagating malware is high. These groups have the resources and time, to launch sophisticated attacks that

leverage and exploit complementary technologies such as short-URLs and target advertisements. This chapter investigates if Twitter can be used to spread malware and a formal threat model for Twitter is built that analyzes the threat and feasibility of propagation of malware via Ywitter. Various attack scenarios are presented along with the mathematical analysis of the costs and effects of the attacks.

6.2 Preliminaries

This section first presents the Twitter user model and the specifics regarding the model, it then highlights the vulnerabilities of Twitter. The section finally showcases some of the attacks on Twitter to better motivate the research challenges.

6.2.1 Twitter User Model

The structure of Twitter can be visualized as two distinct entities: a User → Follower model and a # - tag model. The user → follower entity abstracts the dissemination model of information from a user to her followers, other users who "follow" a user. Information dissemination occurs through *tweets* which are Twitter specific messages. The tweets, which are a broadcast to the world, have a limit of 140 characters. By using the string *@username* at specific positions, a tweet is classified either as a direct message or a "mention" thus bringing the specific message to the attention of the user. The tweets of a user are available only to her followers and cannot be accessed by anyone else. However, if a follower of the original user "retweets" the tweet, the followers of the said follower gain access to the tweet. Figure 6.1 depicts this model and process, where a user's tweets are accessible to her followers only. When a tweet is retweeted by follower number seven, her followers (indicated in red dashed circles) gain access to the user's tweet. On being retweeted by a follower (numbered 1), users who are not followers of the original user also gain access to the tweet. As we can see, the User → Follower model of Twitter has a tree structure, where the information flow occurs down the tree. It is important to also note that, unlike other social networks, the relationship between a user and her follower can be asymmetrical.

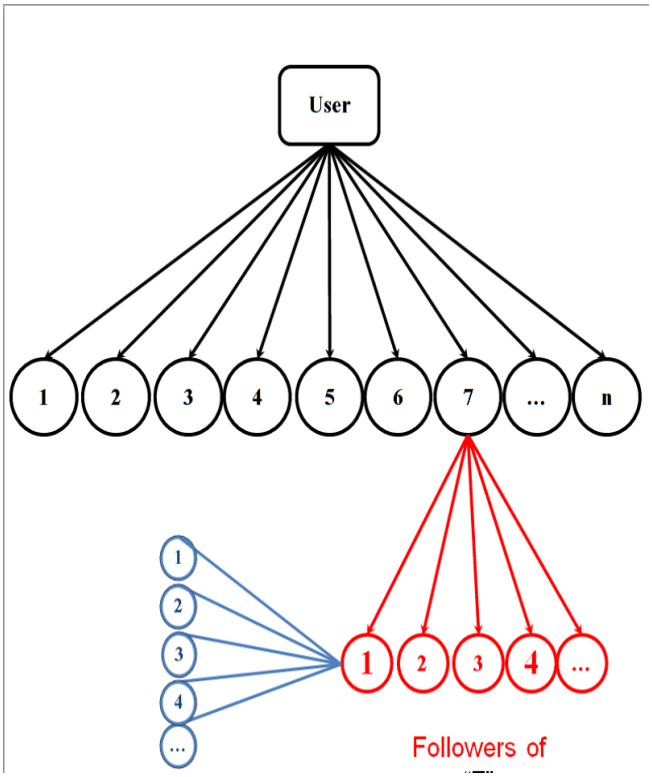

Figure 6.1: Twitter Structure

Specifically, when a user gains a follower, they both do not automatically follow each other, thus a user does not necessarily gain access to *all* the tweets of their followers. Tweets can also be used to broadcast information about specific topics by appending "#-tags" to it. These #-tags are used in determining "trending topics list," which describes the topics that are generating most interest (in a geographic location). These # -tags have been extensively used in market research, disseminating political opinions and obtaining current news. Many Twitter users actively use and follow these #-tags for communication and networking. The #-tag entity comprises of such users. It is important to note that any user of Twitter falls either into the User $\rightarrow$ Follower model and/or a # - tag model. This also includes users who are not following anyone or any *trending* topic. The User $\rightarrow$ Follower and # - tag entities make Twitter a unique model from a security/privacy perspective, since they provides two avenues for a miscreant to leverage. Specifically, the #-tag model provides a miscreant an opportunity to

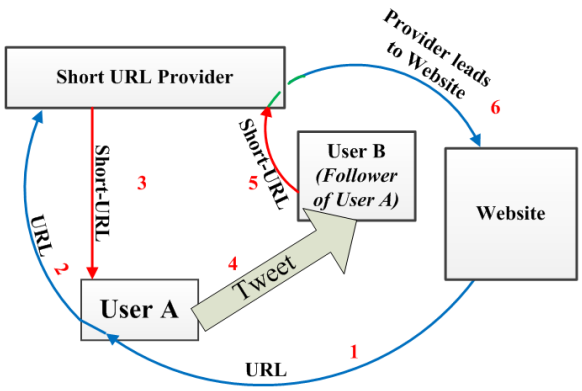

Figure 6.2: How short URLs work in Twitter (numbers depict sequence of operation)

attack a network of user-follower entities which is spatially isolated from other networks of user-follower entities.

6.2.2 Twitter Vulnerabilities

Twitter like various other online social networks, inherently possesses the risks of some miscreants using the medium to share malicious ideas, executables and worms. Personal information can be gleaned through Twitter conversations, that a malicious user can leverage into social engineering kind of attacks Siegel [2009]. The threat of attacks on Twitter can be realistic since these attacks can misuse the trust between users and combine it with other vulnerabilities such as the strict character limit for tweets. Due to the severe limitation on tweets lengths, users use short-Universal Resource Locators in tweets instead of standard URLs. Short-URLs are normal URLs that are encoded into URLs with fewer characters, and can thus be used in tweets. However, short-URLs have some inherent issues of concern. First, some services encode the same input URL into different (unique) short-URLs for different users. Second, unlike traditional systems, a user cannot follow the target of the short-URL (by hovering their mouse over the URL). The short-URL providers such as bit.ly Bit or tiny URL tin services are required to decode them. Figure 6.2 shows the process of using an URL in a tweet. The dashed blue arrows in the figure depict the use of the normal URL, whereas the solid red arrows depict the use of the short-URL. As can be seen

from the figure, a user has a very limited knowledge of the target of the short-URLs. The encoding of URLs is a method of obfuscating information, which can be exploited into tricking unwilling users to download/spread malicious software without their knowledge. Attacks on Twitter can be of different forms, as described in the aforementioned section. In this dissertation, attacks that are targeted on Twitter(its infrastructure or availability) are not investigated, rather the ones that are on the users of Twitter. The study precludes attacks such as spamming of tweets that aims at overloading Twitter's servers. As explained before, the emergence of PPI could make Twitter users an attractive scenario for spreading malware.

6.2.3 Attacks on Twitter

Twitter has been under various attacks ever since its conception. The attacks launched on Twitter and its users have not only become more complex, but due to the permeance of Twitter in the social and cultural context of society, have also been able to target many users. Figure 6.3 shows a timeline of attacks on Twitter and its users from various media reports. While the details of the attacks are not clear/available, it is obvious that miscreants view Twitter as a viable avenue for attacks. It is important to note that the attackers on Twitter have leveraged not only on the specifics of Twitter but also on the vulnerabilities of the Internet. Figure 6.3 showcases how vulnerable Twitter is as an attack avenue for malicious activities such as propagation of malware, especially in the case of zero-day attacks.

6.2.4 Attacks on Twitter

Twitter has been under various attacks ever since its conception. The attacks launched on Twitter and its users have not only become more complex, but due to the permeance of Twitter in the social and cultural context of society, have also been able to target many users. Figure 6.3 shows a timeline of attacks on Twitter and its users from various media reports. While the details of the attacks are not clear/available, it is obvious that miscreants view Twitter as a viable avenue for attacks. It is important to note that the attackers on Twitter have leveraged not

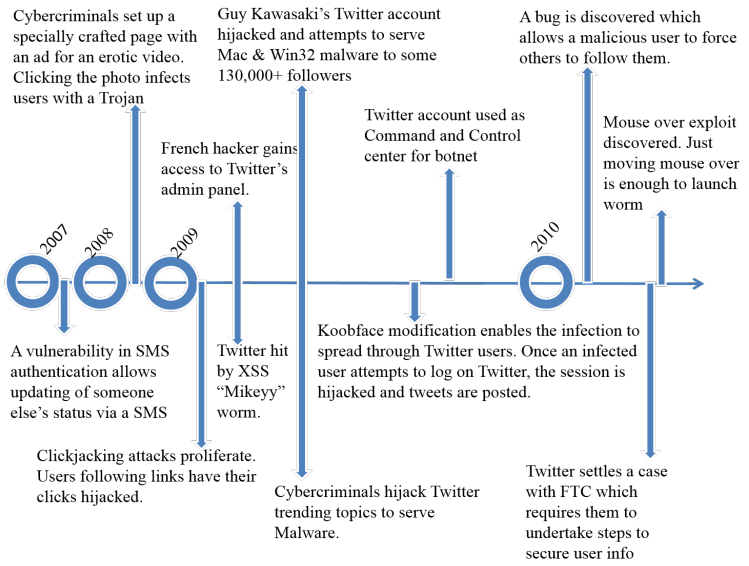

Figure 6.3: Timeline of attacks on Twitter

only on the specifics of Twitter but also on the vulnerabilities of the Internet. Some noteworthy attacks on Twitter are the clickjacking attacks which used Guy Kawaski's Twitter account to serve malware to her followers as well as some attacks that helped in spreading worms using Twitter and its tweets. Figure 6.3 showcases how vulnerable Twitter is as an attack avenue for malicious activities such as propagation of malware, especially in the case of zero-day attacks. Thus, it is important to understand the impact of malware propagation using Twitter, as is done in the following section.

6.3 Analyzing the Impact of Malware Propagation using Twitter

Epidemic theory aims to measure how infections spread and assessing the risk it presents to a population. Epidemic theory builds on parameters such as the number of people who are already infected, people who are exposed, the spread rate of the infection, etc. Epidemic models [17] have been used to model malware spread in a network [5, 6].

Two popular models in epidemic theory are the Susceptible-Infected-Recovered (SIR) and Susceptible-Infection-Susceptible (SIS) models. In SIR model, there is a chance for an infected person to recover from the infection, whereas in the SIS model, the infected person after what is known as the incubation period becomes susceptible to infection again. In this model the more generic SIR model is used, to show that once infected, a user can no longer become a part to infect more people. This would hold true to a large extent in most cases without loss of generality. Of course if an infected user were to get many more followers in this duration, this supposition would not hold true, but it can be safely assumed that the chances of it are minimal. In the SIR model, $S(t)$, $I(t)$ and $R(t)$ denote the number of susceptible, infected and recovered nodes at time t, respectively. Every member of the population belongs to one of these groups. Thus, if $N(t)$ is the total population to be considered then,

$$N(t) = S(t) + R(t) + I(t) \tag{6.1}$$

Here, the standard convention of denoting the infection rate as β and the recovery rate as γ is followed. The recovery rate denotes the removal rate of the infected users from the number of infected users. Since Twitter's method of disseminating information is a broadcast it is assumed that each of the susceptible users (the followers of an infected user) can get in contact with the infectious members and thus get infected. Given these criteria, the equations for the rate of change of susceptible, infected, and recovered members, respectively are:

$$\frac{dS(t)}{dt} = -\beta \times S(t) \times I(t) \tag{6.2}$$

$$\frac{dI(t)}{dt} = \beta \times S(t) \times I(t) - \gamma \times I(t) \tag{6.3}$$

$$\frac{dR(t)}{dt} = \gamma \times I(t) \tag{6.4}$$

The infection rate β denotes the probabilistic rate at which an infected or malicious user broadcasts her tweet containing a link. It is to be noted that based on the attack model of this dissertation, each tweet is also appended with the "#-tag" that is trending at that point. This considers those people who are not in

the network of the followers of the infected user(s) and yet have a (probabilistic) chance of getting the infection by clicking on the link in the malicious tweets. Thus,

$$\text{Number of Infected users} = \text{Infected Followers} + \text{Infected by "\#-tags"} \quad (6.5)$$

However, not all users are equally susceptible; the degree of susceptibility depends on the average degree of connectivity, the rate of trending topic, the probability of malware infection, and the probability of a link being clicked.

Similarly, if the number of Twitter users are considered, they can be divided into two categories - users who are in some way connected to an infected user(N_1) and users who are not connected to the infected users but are following the trend the malware uses via "#-tags"(N_2). Thus, $N = N_1 + N_2$ (where N_1 and N_2 are assumed to be large integers). In this analysis, a no recovery model is assumed, i.e., once infected, the users are compromised and cannot be recovered. This assumption is justified since it eliminates loops between infected $\rightarrow$ recovered $\rightarrow$ susceptible$\rightarrow$ infected and discounts those compromised once, from getting compromised again. Now in this particular model e a non-homogenous mixing is considered, since only followers of a particular user (infected) have the potential to get infected themselves. Thus, users that are outside the particular network are inoperative and cannot get infected or spread the infection. We can visualize this as a circular region of infected nodes, centered around the source user, which grows with time as the infection spreads based on the tweet broadcasts as shown in Figure 6.4.

Since a "no recovery" model is considered, $R(t) = 0$ and $\gamma = 0$ in equations 6.2, 6.3 and 6.4 above. Consequently, if left undetected and unchecked, the infected users will infect all the susceptible users. The number of infected users $I'(t)$ that lie in the network of an infected user is given by

$$I'(t) = I(t) - \sigma\pi(r(t) - 1)^2 \quad (6.6)$$

where σ denotes the density of followers based on time and geographic locations [Rao and Nagpal 2011a;b] and $r(t)$ denotes the radius of the circle that contains the infected users. The parameter $r(t)$ is based on considerations such

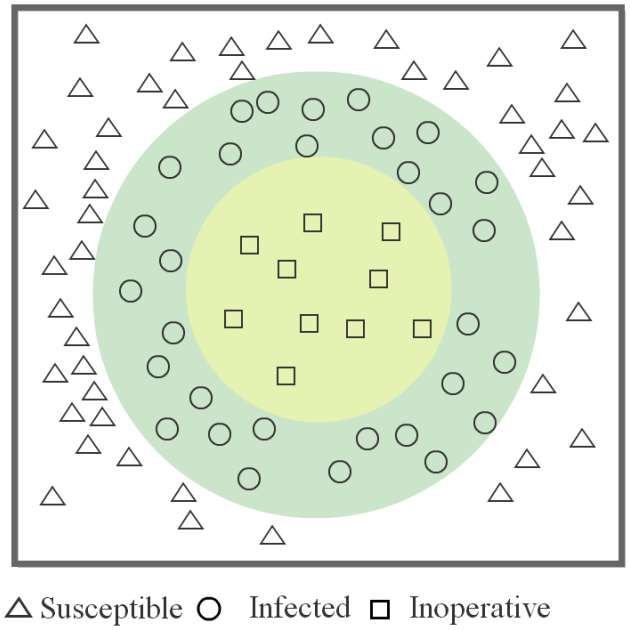

Figure 6.4: The spread of malware to other users/followers

as time difference, geographic location, interest in a particular user's tweets, and that followers of an infected user (who are now infected) can be placed in a circular strip of thickness $r(t)$. This consideration is validated by the practicality of the assumptions we have made so far and the fact that while distance may not mean much in the Internet, time zones and time differences are a factor that determines how tweets are followed or get re-tweeted. Based on this assumption, it is clear that $\sigma\pi r(t)^2 = I(t)$ and $I'(t) = (2\sqrt{\sigma\pi}) \cdot \sqrt{I(t)} - \sigma\pi$. Now $2\sqrt{\sigma\pi}$ is a proportionality constant that is denoted as ε and thus it can be stated without loss of generality that

$$I'(t) = \varepsilon \cdot \sqrt{I(t)}$$

It is important to note that, the average degree of followers of a user η plays an important role and that each user in $I'(t)$ is able to communicate only with η followers. The parameter η depends on the activity of a user, popularity and the interest generated by his/her tweets. Obviously, a user with a high degree of followers can spread the infection faster, than one with lesser number of followers. As mentioned above, the probability of a follower clicking on a link is the highest

determining factor in the follower getting infected which we denote as τ. The analysis so far only considers the user→ model, i.e., users infecting followers. Since it is considered to be N_1 users, the relationships between the susceptible and infected users can be rewritten as:

$$N_1(t) = S_1(t) + I_1(t) \tag{6.7}$$

Thus, the equations for rate of change of infected and susceptible users become:

$$\frac{dI_1(t)}{dt} = \beta \cdot \varepsilon \cdot \tau \cdot \sqrt{I_1}\frac{(N_1 - I_1)}{N_1} \times \eta \tag{6.8}$$

$$\frac{dS_1(t)}{dt} = -\beta \cdot \varepsilon \cdot \tau \cdot \sqrt{I_1}\frac{(N_1 - I_1)}{N_1} \times \eta \tag{6.9}$$

Substituting, $U = 1/\sqrt{I_1}$, the first equation can be simplified into the following:

$$\frac{dU}{U^2 - \frac{1}{N_1}} = -\frac{\beta \cdot \varepsilon \cdot \eta \cdot \tau}{2}dt \tag{6.10}$$

which after integration on both sides and applying the boundary condition $I(0) = 1$, i.e., initially only one node was compromised, leads to the following:

$$I_1(t) = N_1 \times \left(\frac{2}{1 + \left(\frac{N_1 - 1}{N_1 + 1}\right) \cdot e^{-\frac{\beta \cdot \varepsilon \cdot \tau}{\sqrt{N_1}} \cdot t}} - 1 \right)^2 \tag{6.11}$$

Similarly, if the rate of infection spread is based on "#-tags." the general form of the equation remains the same as the user→ follower model, except that the probability of infection is also determined on the rate of the trend and the probability of a (non-compromised) user clicking the link denoted by ϕ . Thus combining the two factors the total number of infected users as time t are

$$I(t) = N_1 \times \left(\frac{2}{1 + \left(\frac{N_1 - 1}{N_1 + 1}\right) \cdot e^{-\frac{\beta \cdot \varepsilon \cdot \tau}{\sqrt{N_1}} \cdot t}} - 1 \right)^2 + N_2 \times \left(\frac{2}{1 + \left(\frac{N_2 - 1}{N_2 + 1}\right) \cdot e^{-\frac{\beta \cdot \varepsilon \cdot \tau}{\sqrt{N_2}} \cdot t}} - 1 \right)^2 \tag{6.12}$$

It is important to note that as $t \to \infty$, the term $1 + \left(\frac{N_1 - 1}{N_1 + 1}\right)^{\left[\frac{\beta \cdot \varepsilon \cdot \tau}{e \cdot \sqrt{N_1 t}}\right]}$ becomes 1. Thus, the entire population becomes infected, i.e.,

$$I(t) = N_1 + N_2 \text{ as } t \to \infty \tag{6.13}$$

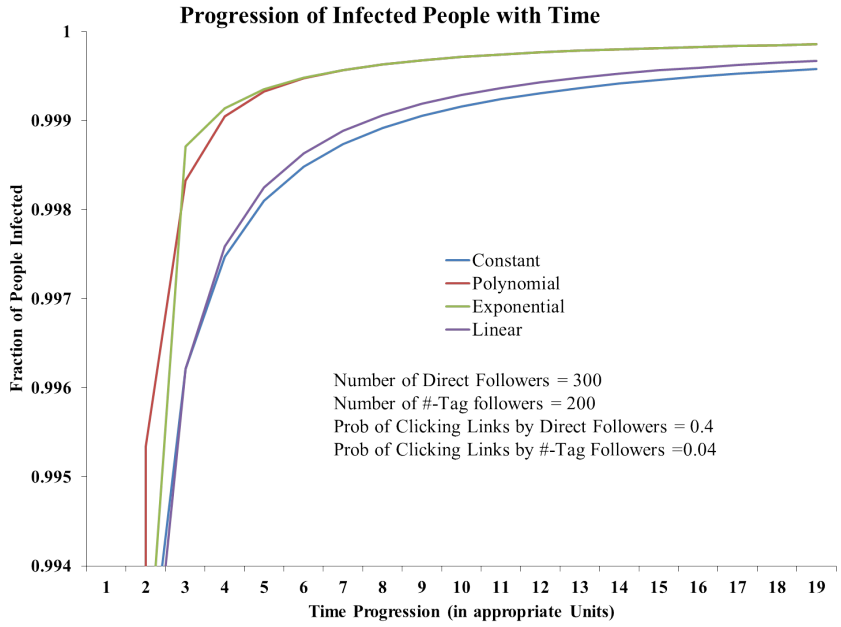

Figure 6.5: Progression of Infection

6.4 Results

Figures 6.5, 6.6 and 6.7 show the plots of the fractions of infected users as time progresses for different number of followers and different probabilities of clicking links. The plots are based on equation 6.12 for different data trend rates as shown in the figures. For simplicity it is assumed that all users have the same number of followers. The x-axis represents the progression of time while the y-axis represents the fraction of infected people. As can be seen from the plots the fraction of infected users increases directly based on the number of followers and

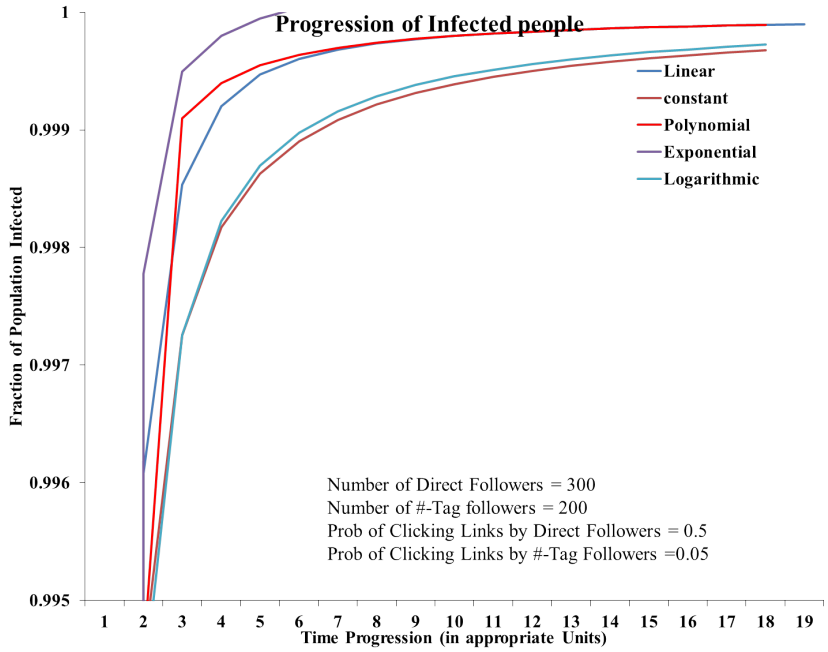

Figure 6.6: Progression of Infection-II

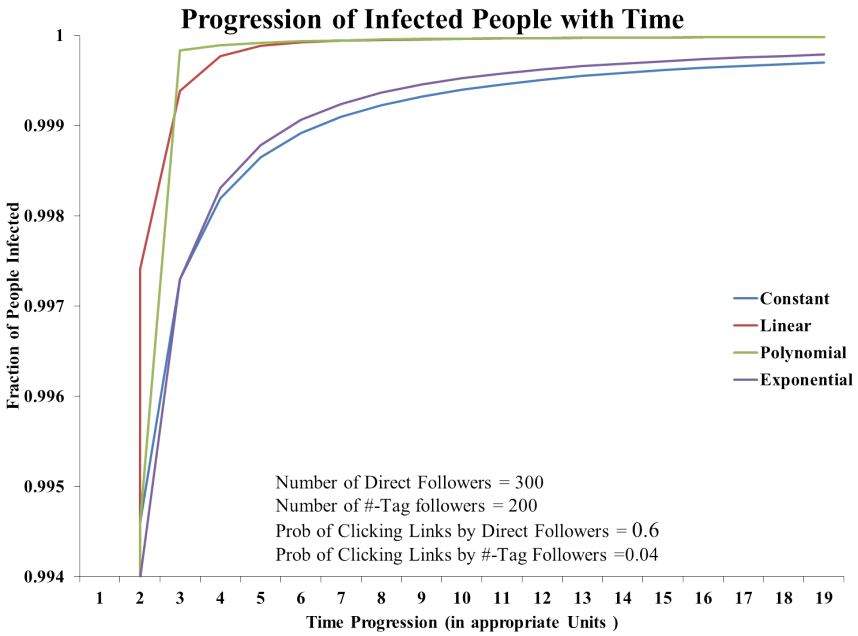

Figure 6.7: Progression of Infection-III

the probability of them clicking the links. Further, the data trend rate also plays as important part, as can be observed from the graphs. Exponential data trend not only represents the spread of malware faster, but also has the highest initial amount of infection. For the constant data trend, the spread of infection is very low as is the initial amount of infected users.

The analysis so far, assumes that a user of Twitter has already been infected and based on this the quantitative analysis of how the infection would spread in Twitter is provided . However, the question of how can one infect the "first" user arises. The following section presents the scenarios and threat assessment of how one can infect even the first user and how one can spread the malware infection.

6.5 Threat Model of Twitter for Spreading Malware

The following section presents the methodology of infecting one user in Twitter. Next, the analysis of this simple attack is presented, before proceeding to more complicated and close to real world attacks that a miscreant can use to infect Twitter.

6.5.1 A Common Attack Methodology

To spread malware using Twitter and its users, any miscreant would have to first encode the malware site as a short-URL. Now to disseminate this information she would have two approaches – a) Use as many *@username* in her tweets and hope some users click on the link or b) Compromise and control a user account and then post the tweet to her followers. Figure 6.8 depicts the two common attack methodologies. The upper part of the figure, depicts the methodology of sending directed messages to users while the lower part of the figure shows the process of using a compromised account to send tweets to followers. A briefly analysis of the pros and cons of both these methods is presented shortly. In its simplest form, this attack would be similar to the *koobface botnet* attack which misled users into going to a malware site and then forced them to download the malware under the pretence of updating their flash player or other software. However, this attack

depends on the probability of the infection of the malware and the probability of a user clicking on a link.

The attack introduced above is a simplistic one that has the potential of infecting many users at the same time. However, from a practical consideration, there are certain aspects that need to be considered for this attack.

6.5.1.1 Analysis of the attack

Of the two attack scenarios presented, the attack in the first scenario while being low cost in terms of resources, has two important obstacles. First, Twitter has strict spamming standards; a user is termed as a spammer if either a user follows too many users or (and) posts many *@username* posts. Even if a miscreant is able to get around this obstacle, the other issue is that the probability of a user clicking on a @username from an unknown user (and in an unknown context) will be low. In the second attack scenario, a miscreant will have to spend considerable time and resources to compromise and control an account. However, once in control of a user account, the miscreant can then proceed with the attack as mentioned. The probability of other users (followers of the user) clicking on a link in this case is higher than the case described earlier, simply because the trust between a follower and a user is high due to their interactions over time. Another aspect to consider is that the propagation/installation of malware depends on the degree of infectivity of the malware, which can be affected by devices used by the victims, operating systems, frequency of patching operating systems, etc. The attacks described here, may not infect many users due to the reasons explained above. Further, this attack does not really consider or leverage the User $\rightarrow$ Follower model, which would allow a miscreant to reach deeper into the network. The following section presents an advanced version of this attack which follows the principle of the common attack but also leverages the Twitter structure as well as its analyses.

6.5.2 An Advanced Self-Propagating Attack

Based on the premises of the above simple attack one could conceive an advanced attack that leverages the User $\rightarrow$ Follower model of Twitter. The advanced self-

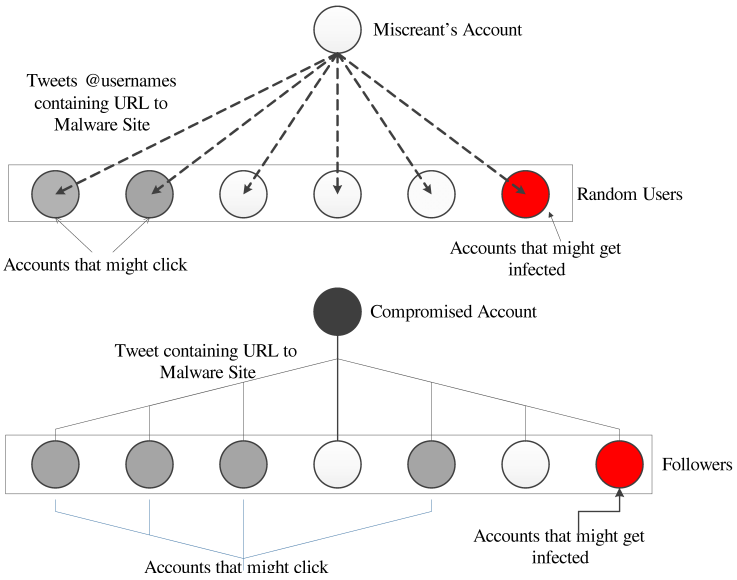

Figure 6.8: Attack scenarios depicting the simple attack

Table 6.1: Notation Summary

Model Parameter	**Description**
d	Depth of the tree
η	Avg. Number of Followers
ρ_{click}	Prob. of user clicking
ρ_{retweet}	Prob. of user re-tweeting
ρ_{malware}	Malware Infectivity
N	Number of Susceptible Users
$N_{\text{infection}}$	Number of Infected Users

propagating attack uses the *clickjacking* technology. There are two important considerations to take into account while designing this attack. First, the attack needs to exploit the inherent trust between a user and a follower. As explained above a link is more likely to be clicked by a follower of a user than by another non-following user. Second, the attacker needs to consider that in Twitter, information (in this specific case the malicious short-URL) can only propagate down the tree if it is retweeted by the followers. Thus, the advanced attack would need to involve *clickjacking* such that the tweet retweets itself whenever a user of Twitter (follower) clicks on the link. The clickjacking attack can also exploit the weakness of short-URL providers which encodes a new short-URL for different users. This attack has the benefits of propagating down the Twitter tree, with the additional benefits of making it difficult for Twitter to analyze the different short-URLs due to the amount of information generated and traversing through the network.

6.5.2.1 Analysis of the Advanced Attack

Let us assume a Twitter tree structure as shown in Fig. 6.1, with a depth $d+1$ and the average number of followers for each user as η. Let each follower with a probability ρ_{click} clicks on a link in a tweet and with a probability $\rho_{retweet}$, retweets a link. For the User $\rightarrow$ Follower model we can assume that $\rho_{click} \geq \rho_{retweet}$ since not all followers might retweet it. Now, the equation for the total number of users (N_1) in the tree of depth $d+1$, who would have see a benign link in the tweet would be

$$N_1 = \sum_{i=1}^{d} \left({\rho_{click}}^i \times {\rho_{retweet}}^{i-1} \times \eta^i \right) + 1 \tag{6.14}$$

where the term "1" is for the main user who starts propagating/tweeting the particular tweet. Similarly, considering the advanced attack, where retweeting happens automatically, the total number of people who can see the link would be:

$$N_2 = \sum_{i=0}^{\mathrm{d}} \left({\rho_{click}}^i \times \eta^i \right) \tag{6.15}$$

where N_2 describes the number of users who see the malicious link. Here, $N_2 > N_1$ since retweeting happens automatically in the advanced attack and would thus go

further down the Twitter tree structure. Similarly, the calculation for N_2 starts from 0 since, there is a chance that no one sees the tweet, unlike in the calculation of N_1 where at least "1" user has to start the tweet. It is important to notice that as the depth of the tree increases (i.e., d $\rightarrow \infty$), the number of users who can see the malicious link will also increase and $N_2 \rightarrow N$, where N represents all the users of Twitter. Thus, the attack would theoretically encompass all the users of Twitter; which is unlikely. Another consideration is that even in this case, N_2 or N, does not denote *the number of infections*, rather it is a measure of the number of users who are susceptible to the malware. The total number of infections depends on the probability of infection of the malware. Simply put,

$$\mathrm{N_{infection}} = \rho_{malware} \times \mathrm{N} \tag{6.16}$$

where $N_{infection}$ is the total number of infections and $\rho_{malware}$ represents the probability of infection of a malware.

6.6 A Complex Indirect Attack

The analysis and attacks discussed so far has assumed that a miscreant either randomly tweets to other users or compromises and controls a genuine user account and then engages in the attack. The success of the attack depends on the ability of a miscreant to effectively and efficiently compromise user accounts which is a cost intensive process. Further, while Twitter may be a new medium, the authentication mechanism of usernames and passwords is not, which means that if users use strong passwords, it might be difficult to crack. This means that the cost is going to rise exponentially, when the miscreant tries to take over more than one account to ensure a high probability of success. Further, a miscreant may get diminishing returns for her investment due to a number of dependencies (probabilistic) for successful infection. However, this does not imply a low level of risk for attack on Twitter. An important aspect for consideration is that the compromise of an account is not limited to the purview of Twitter; the clickjacking attack in principle can be modified to take place even outside the purview of Twitter. A more plausible attack scenario, is where a Twitter user is surfing

public websites which also allow users to enter links to other websites, such as blogs and news sites. Such sites provide an avenue for a miscreant to insert malicious short-URLs, and clicking on the links would instigate the advanced attack if the victim is also a Twitter user. Given this scenario, it can be assumed that the miscreant is inserting links in between conversations people are having in the comments section of a popular news article. Assuming that any user of the website will randomly click on a malicious link (given that a user will click on a link on the website) the following factors affect the probability of the user clicking on the malicious link:

1. A posting strategy by miscreants in relation to the number of posts at a given time such that they control the majority of the posts.

2. The probability of a miscreant posting the malicious links at any given time frame.

3. The user's probability of clicking on a link.

The first factor represents the number of posts that are occurring by other users. In relation to this, a miscreant also needs to input her links so that the malicious links can be seen by users and thus have a *chance* of being clicked on. The second factor represents the probability of posting links to help miscreants circumvent detection techniques implemented against spammers. Further, there could be many such spammers who also are attacking such websites; a miscreant with the aim of spreading malware may also be competing with other PPIs or PPCs. This probability helps the miscreant to post links based on her discretion (if done manually). Finally, these strategies do not make any sense, if no users click on any of the links. The following assumptions are made for all cases:

1. There are some users who are reading/clicking/entering links (benign or malicious) on a certain website. The total number of such users is denoted by $\eta_{website}$ and the probability of a user clicking on a link is ρ_{web_click}.

2. The number of posts being entered by users other than the miscreant is some function of the number of users who are on a particular website, i.e., $\Phi(\eta_{website})$.

3. Similarly, the number of posts (links) by the miscreant is also based on the number of users who are on a particular site, i.e., $\Psi(\eta_{website})$.

6.6.1 Analyzing the Complex Attack Scenario

The complex attack scenario requires a miscreant to basically exploit social aspects to increase the probability of spreading malware, by utilizing other websites to launch her attack. However due to the fact that even posting malicious links costs the miscreant and there is a probability of a miscreant being detected or termed as a "spammer", we have two scenarios, namely,

1. Miscreant posts malicious links *all* the time, such as using a script or manually inserting links.

2. Miscreant posts malicious links probabilistically.

Both scenarios are analyzed in the following section to find which scenario best suits a miscrean

6.6.1.1 Posting Malicious Links all the Time

In this scenario a miscreant posts a malicious link (mal-link) every time another innocent/malicious post is made. Given that a link was clicked by a user, the probability of the link being malicious is

$$\Pr(\text{Click on a mal link}|\text{Clicked on a link}) = \frac{\Psi\left(\eta_{website}\right)}{\Psi\left(\eta_{website}\right) + \Phi\left(\eta_{website}\right)} \tag{6.17}$$

Now if assumed that $\Psi(\eta_{website}) = \Phi(\eta_{website}) + \varepsilon$, such that $\varepsilon > 0$, then the above equation can be rewritten as

$$\rho_{link} \approx \frac{1}{2} + \varepsilon' \tag{6.18}$$

where ρ_{link} is the conditional probability Pr(click on a mal_link|clicked on a link) and $0 < \varepsilon' < \frac{1}{2}$. Now the probability of the malicious link being clicked is $1 -$ Probability that the link is not clicked; i.e., a user clicked on a link, but the

link was not the malicious link or the user did not click at all. If the attacks were repeated over x trials, then the equation can be written as:

$$\Pr(\text{click}_{mal\_link}) = 1 - \Big[\Pr(\text{click})\times \\ (1 - \Pr(\text{click mal\_link}|\text{clicked a link})) + (1 - \Pr(\text{click}))\Big]^{x} \tag{6.19}$$

Using the definitions from Eq. (6.17) and Eq. (6.18), the Probability of clicking on a malicious link simplifies to,

$$\Pr(\text{click}_{mal\_link}) = 1 - \Big[\rho_{web\_click} \times (1 - \rho_{link}) + \ (1 - \rho_{web\_click})\Big]^{x}$$

For one instance of the attack (i.e., $x = 1$), the probability of clicking a malicious link would simply be,

$$\Pr(\text{click}_{mal\_link}) = \rho_{web\_click} \times \rho_{link} \tag{6.20}$$

While the probability of some user clicking on the malicious link increases with more number of trials, a miscreant might not want to post links all the time due to the non-zero cost (in terms of both time and resources) involved in posting the link. Further, there is always the risk of being termed as a spammer. To avoid these situations, a miscreant might need to post the link based on a probability (denoted as q). The analysis of this scenario follows next.

6.6.1.2 Probabilistically Posting Links

A miscreant can either post a link or not post a link, in a probabilistic sense. Then the only aspect that changes from the previous scenarios, will be the conditional probability in Eq. (6.18). Thus, the probability now becomes:

$$\rho_{link} = \big(\frac{1}{2} \approx \varepsilon'\big) \times q \tag{6.21}$$

where q is the probability with which the miscreant posts her links. This new probability can be replaced in Eq. (6.20) to get the probability of a user clicking

on a malicious link.

$$\rho_{link} = \left(\frac{1}{2} \approx \varepsilon'\right) \times q \tag{6.22}$$

where q is the probability with which the miscreant posts her links. This new probability can be replaced in equation (6.20) to get the probability of a user clicking on a malicious link. The analysis so far presents the probability of a user clicking on a malicious link using different malicious link insertion strategies. The result of clicking the malicious link is that the followers of the users (who we now call the *root*) become susceptible to the malware. Thus, the number of users susceptible to malware by clicking on the malicious link, by any user is a function of:

1. The probability of a user clicking on the root, Pr(click mal_link), which will be denoted as $P_{\text{mal\_link}}$.

2. The number of people who are active on the website, the miscreant is targeting.

3. The depth of the User $\rightarrow$ Follower model, i.e., the total number of followers of the said user and their followers.

Hence, the equation for the number of susceptible users to propagate the malware is:

$$N_{\text{susceptible}} = P_{\text{mal\_link}} \times \eta_{\text{website}} \times \underbrace{\sum_{i=0}^{d} (\rho_{\text{click\_Twitter}} \times \eta_{\text{Twitter}})^i}_{\text{A}}$$

where the term A is a form of Eq. (6.15) from Section 6.5.2; the terms $\rho_{\text{click\_Twitter}}$ and $\eta_{Twitter}$ represent the probability of clicking links in tweets (different from clicking links in websites, i.e., $\rho_{\text{click\_Twitter}} \neq \rho_{web\_click}$) and the average number of followers per user in the User $\rightarrow$ Follower model, respectively. Similarly, the number of infected users can be written as

$$\text{N}_{\text{infection}} = \rho_{malware} \times \text{N}_{\text{susceptible}}$$

At this point, it must be clarified that based on this attack and its analysis, a miscreant can have potentially more strategies (at a higher or lower cost) by tweaking

the parameters. However, the overall strategy of probabilistically posting malicious links still remains the same. More details are provided in Section 6.8.

6.7 Extension to Hash Tags

The analysis presented so far is based on attacks that primarily target the User → Follower model of Twitter. However, one aspect of Twitter that is unique is the # - tag model which provides a miscreant the ability to infect users that are not connected in any way to an infected network. This model can be exploited to propagate malware deeper and to newer networks. The attack to target this particular model can be constructed in conjunction with the attack that targets the User → Follower model by simply appending an #-tag to the tweets. This makes the tweet visible to those users of Twitter, who follow only #-tags. The attack then behaves as described earlier, targeting the followers of this particular user who belong to a different network. The analysis of this attack is also similar to the analysis of miscreants inserting links into websites as discussed in Section 6.6. The strategies of posting the tweets also remain largely the same with the exception of the following two choices:

1. Appending a #-tag that is already trending.

2. Appending a new #-tag that the miscreant creates.

Both these choices directly affect the probability of the malicious link being clicked (which is now encoded in a tweet) and thus the number of susceptible users.

6.7.1 Analysis of Attack in the #-Tag Model

In the #-tag model, the factors that affect the probability of a malicious link being clicked in Eq. (6.17) and Eq. (6.20) are:

- The number of users following a particular #-tag ($\eta_{\#\text{-tag}}$) and their probabilities of clicking on a link in a tweet with #-tags ($\rho_{\text{click }\#\text{-tag}}$, where $\rho_{\text{click }\#\text{-tag}} \leq \rho_{web\_click}$).

- The number of posts that are being generated by other users that are appended with #-tags ($\Phi_{\#\text{-tag}}$).

6.7.1.1 Miscreant Enters Trending #-Tag

If a miscreant starts using an already trending #-tag, the factors that affect the probability of the malicious link being clicked are the number of people who are following the trend and the rate with which the posts are made. Further, there exists the cost of analyzing topics that would maximize the chance of people clicking on links. Similarly, the miscreant also has to evaluate the duration that a topic may remain trending. For example, a trending topic that is related to local news will have a smaller group of people following as compared to national level topics or a global level topic. At the same time, a local topic might have a higher chance of remaining a trending topic for a longer duration than a national or global topic. Thus, the probability of a malicious link being clicked is:

$$P_{\text{mal\_link \#-tag}} = 1 - \left[(\rho_{\text{click\_\#-tag}} \times (1 - \rho_{\text{link\_\#-tag}}) + (1 - \rho_{\text{click \#-tag}})\right]^{x} \quad (6.23)$$

where x is the number of retries and $\rho_{\text{link\_\#-tag}} = \frac{1}{2} + \varepsilon'$ if the miscreant is appending the #-tag all the time or $\rho_{\text{link\_\#-tag}} = \left(\frac{1}{2} + \varepsilon'\right) \times q$ if the miscreant is appending the #-tag based on a probability q.

6.7.1.2 Miscreant Creates Her Own Trending #-tag

Similar to Section 6.7.1.1, if the miscreant decides to create her own #-tag and appends it to all the tweets, the first factor she will have to consider is the probability of any user being interested in this topic and clicking on the link. However, a greater consideration will be the time it takes for this topic to actually become a trending topic. Simply put, the term $\rho_{link\_\#-tag}$ which is a function of the number of users following a given #-tag will now have to account for *all* the users of Twitter, since the #-tag will be competing at a global scale. This means that the number of posts that need to be generated will simply be too large for the miscreant to even have a chance for a user to click on which will also increase her cost by a large amount.

Thus, the case of a miscreant creating her own #-tag is too costly in terms of both time and resources to be considered by a miscreant, although other variations of the attack may still exist. However, those variations are beyond the scope of this dissertation.

6.8 Cost Analysis and Discussions

So far, all the conceptualized attacks on Twitter have been analyzed using probabilistic methods for different strategies, which gives a probabilistic measure of the degree of success and penetration into the Twitter network. While all these attacks have been mathematically modeled the identification of aspects/parameters that can be controlled by the miscreant (to increase the chance of her success) is still necessary to analyze the feasibility of these attacks. The following section first identifies the parameters of this model that a miscreant will have to consider/control for an attack and validates the identification by empirically inserting values in the equations and analyzing the results. Next the results from a simulator built using NetLogo Tisue and Wilensky [2004], to simulate the propagation of attack on Twitter is presented. Finally, computational analysis to assess the amount of work for a miscreant to launch an attack on Twitter using the attack scenarios is presented to understand the feasibility of attack on Twitter.

6.8.1 Parameters of Interest to a Miscreant

The models of attack albeit probabilistic, describe the relationships between the various parameters that an attacker has to consider. In the best case scenario, an attacker may execute her attack in a manner that gets the probability $P_{\text{mal\_link}}$ to be as close to 1 as possible, targeting a polynomial number of infections. However, the time taken and the cost of inserting links may make the attack moot and infeasible. Conversely, an attacker may choose a low degree of infection and a low probability of a user clicking on her link, to save cost and detection and repeat the attacks over longer time. Thus, at least at a high level, there is the

inevitable tradeoff between the cost of an attack (in terms of time, resource and implementation of attack) and maximizing the number of susceptible users.

Analyzing the various equations for the different strategies of attack, the following factors can be controlled by a miscreant:

1. The number x of trials.

2. The probability ρ_{link} of the malicious link being clicked, given that a link was clicked by a legitimate user. This has been assumed to be $\frac{1}{2} + \varepsilon'$. However, this depends on the number of posts by both legitimate users (Φ_{website}) and the miscreant (Ψ_{website}).

3. The probability $P_{\text{mal\_link}}$ that a malicious link was clicked.

However, not all factors can be controlled at the same time; a miscreant has to insert a number of posts depending on the number of posts by legitimate users to stay in contention. To try and maximize $P_{\text{mal\_link}}(\text{wanting})$, a miscreant can only do so after a certain number of trials x which is dependent on the probabilities of people clicking the links. The miscreant can also reduce the number of posts per trial to save the cost while lowering the degree of infection. The resulting "amount of work" and "number of susceptible" users under such conditions can only be probabilistically measured.

It is obvious that the success of the attack of a miscreant depends on the legitimate users' probability of clicking links (both malicious and benign). Even with low probabilities it is still quite possible for successful attacks. Figure 6.9, under the assumption of a small probability for users clicking, plots the probabilities of users clicking on a malicious link for different number of trials and different probabilities (q) of miscreants posting links. It can be seen that a miscreant can achieve a high probability of $P_{\text{mal\_link}}$ by increasing the number of trials. Further, for even a small probability of posting links a miscreant can still guarantee a good probability of users clicking on the malicious link which would lower the chance of the miscreant being detected as a spammer.

Similarly, the number of susceptible users depends directly on the depth of the User $\rightarrow$ Follower model and the average number of followers at each depth (η_{Twitter}). Figure 6.10 shows the change in the number of susceptible users for

an increasing number of followers and depths. The probabilities of users clicking links (ρ_{link}) and the number of trials of attack are fixed. The plot shows that even at small depth the attack can still affect a large number of users. Also, the cost of such an attack would be low for a miscreant, since the fixed parameters have been chosen conservatively.

Figure 6.11 provides more insight into the strategies a miscreant can leverage to increase the chance for a successful attack for different depths as well as different probabilities of users clicking links. Here it is to be noted that if the attacker can shape the attack by enticing users to click on links even by a small probability of 0.01, the number of susceptible users increases drastically. Further, by targeting users who have a lower average number of followers but a higher probability of clicking links, a miscreant can save on her cost of attack while still maintaining her target of susceptible users. This also validates the assumptions that successful attacks might be possible even when users do not have a high probability of clicking random links.

The figures above corroborate the identification of the parameters that a miscreant can potentially manipulate to increase her chance of success. The following section first validates the factors identified via simulations and then presents the analysis of the amount of work (cost) required for a successful attack.

6.8.2 Simulation Results

In the previous section, three factors an attacker might be interested in have been identified, based on the attack scenarios. These factors were derived quantitatively from our probabilistic model, where some aspects such as the same number of followers for every user, the same probability of a follower to click on links, etc., might not hold true in the real world. To validate whether these factors would still hold true, our attack scenarios were performed in a simulated platform using the agent-based simulation tool NetLogo. The Twitter simulator is built using the NetLogo programming language. The simulator creates a Twitter like structure, for a user specific number of nodes, and configures them into the User $\rightarrow$ Follower model. This configuration is done probabilistically based on a preferential attachment model Flaxman et al. [2004], where the nodes are likely

to follow some user with more followers or could randomly connect themselves to other nodes.

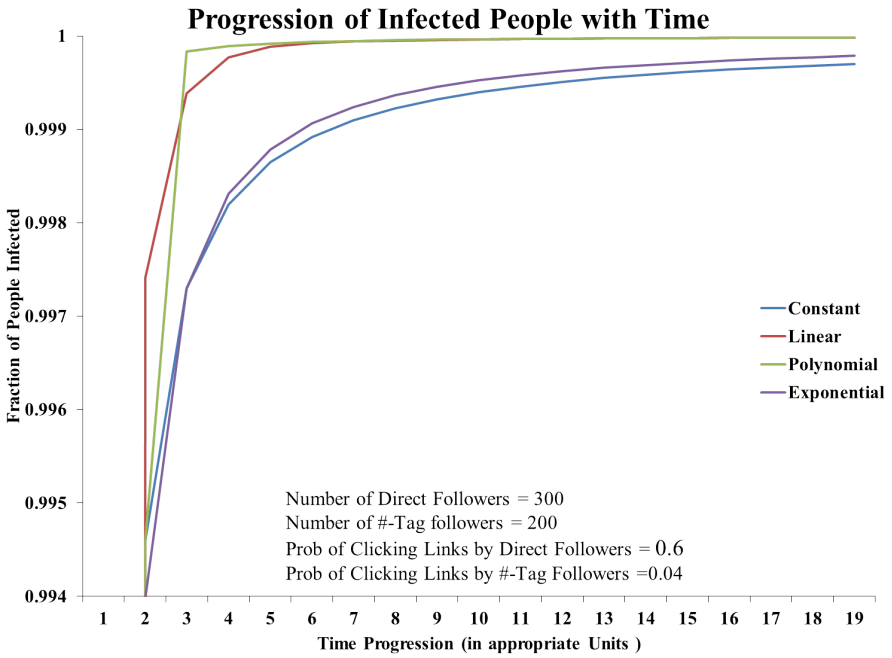

Figure 6.9: The effect of the no. of trials on probability of malicious link being clicked

Experimental Set Up and Procedure. The NetLogo simulator allows configuration of parameters such as number of users, maximum probability of clicking on a link, probability of retweeting, max probability of getting infected, max probability of tweeting new content as well as list of tweets that are viewable for each node. Similarly, one can also configure parameters such as probability of inserting comments on a blog and clicking of links on a blog for *each* node. The latter parameters are used in simulating the complex attack scenario. A separate node (not part of the Twitter structure, termed as a miscreant is used to insert malicious links into the "blog" based on a probability (q). Figure 6.12 shows a screenshot of the simulator along with some set up parameters such as type of network topology, number of users, etc. The circles in the screenshot depict the various users and the connections between them based on the User $\rightarrow$ Follower

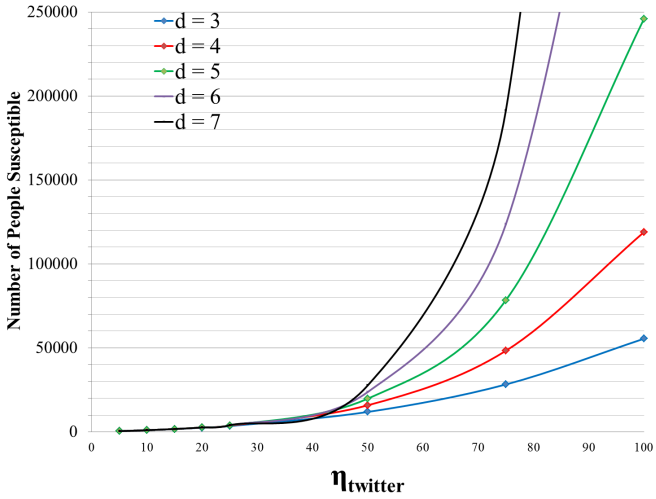

Figure 6.10: The probabilistic estimate of no. of susceptible for different depths

model. On starting the experiment, the simulator randomly cycles through the nodes and based on the probability takes one of the following actions – either creates and tweets new content, simply views a tweet or retweets a tweet (if any are viewable). If a retweet occurs, the simulator will update the viewable tweet list of all of the followers. We also configure the simulator to insert links (based on a probability) if it tweeted new content. Similarly, each node based on a probability, will either insert a comment on the blog, click on a blog entry if it could view it or do nothing related to the blog. To model a real world scenario, the probability of acting on a blog was set to be less than that of Twitter, unless a certain number of comments had been inserted in the blog in consecutive "ticks" of simulation.

Infection and Susceptible. If a node clicks on a link from the miscreant, the simulator randomly generates a number and compares it to the node's probability of infection. If the number generated is higher, then the node is termed to be infected. Subsequently, all further contents from the infected node are deemed to contain malicious links, if they have links inserted. The simulator also ascertains in a similar manner, if a follower node of the infected user has been infected. All simulations are run until 90% of the nodes are infected. All results presented in the following section are averaged over 2000 runs.

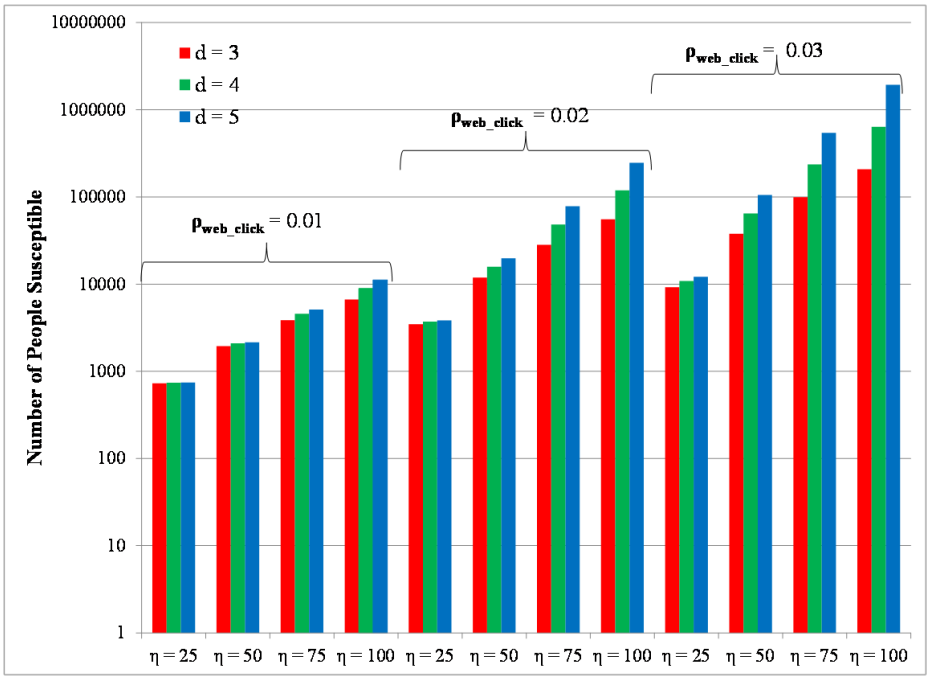

Figure 6.11: The effect of varying the no. of followers and probability of clicking on links on the number of susceptible users

Simulation Results. Figure 6.13 shows the comparison of the theoretical and simulated probabilities of a malicious link for different number of trials when a miscreant has different probabilities of inserting the links in a website or blog. As can be seen from the figure, the simulated values are very close to theoretical calculations with a maximum error of 10%. The simulated and theoretical probabilities of the malicious link are equal when the probability of inserting links is 0.75, thus causing an overlap of lines in the plot. This shows that the chances of a user clicking on a malicious link is affected not only by the probability of inserting malicious links but also by the number of trials the miscreant inserts links.

Figure 6.14 shows the plot of the number of infected users for different click probabilities. From the plot, it can be inferred that once the top level users are infected from the blog there is a significant increase in the number of followers getting infected in a short span of time. This follows our intuition that once the top level users are infected, the malicious link propagates faster through the

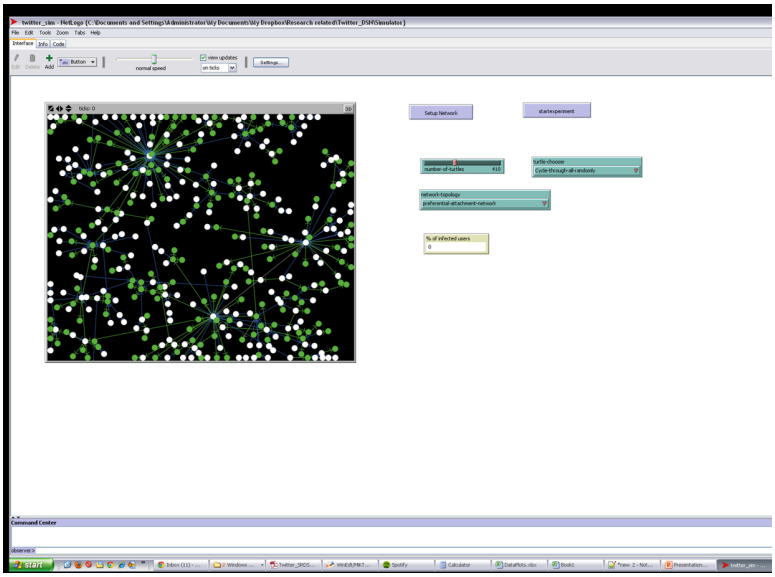

Figure 6.12: Screenshot of Twitter simulator for illustration purposes only

Twitter tree structure. An interesting point to observe is that, even an increase by 0.01 in the clicking probabilities causes a significant increase in the number of infected users. Further, it can be observed that if the click probability is small (0.01), it takes significant time to infect the top level users, thus delaying the infection down the tree.

6.8.3 Cost Analysis

This section presents the cost analysis of launching an attack on Twitter and its users by a miscreant and summarized it in the form of a table. Table 6.2 gives the analysis for the amount of work that occurs for different $P_{\text{mal\_link}}$, assuming different user posting activity (Φ_{website}) and with the miscreant aiming for different probabilities of users clicking the malicious links. The amount of work a miscreant has to perform is the product of the work that she is *forced* to do ($\Psi(\eta_{web})$), along with the work she needs to do to increase her chance of success – the number of trials x. For instance, from the first entry of Table 6.2, if it is assumed that the activity of legitimate users on a website is $log(\eta_{web})$ and the miscreant aims at the probability of any user clicking on her malicious link ($P_{\text{mal\_link}}$) to be close to 1, then by Eq. (6.17), it can be seen that the miscreant is forced to at least match

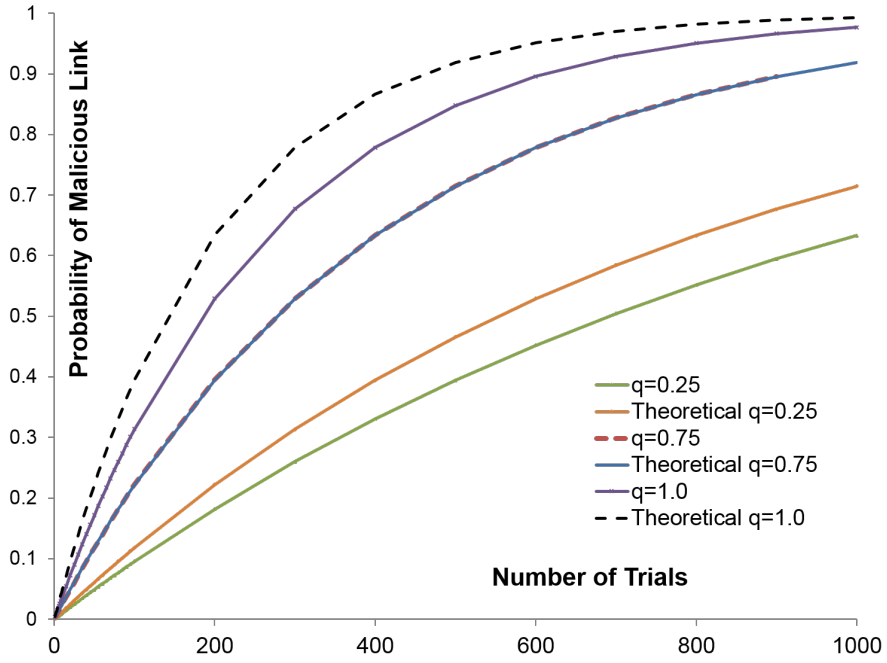

Figure 6.13: Comparison of theoretical and simulated probability of malicious links for various number of trials

the activity so that her link can be seen, i.e., she is also forced to do $log(\eta_{web})$ amount of work. This is because her chance of a successful attack is directly dependent on the activity (or response) from legitimate users. Since it is a factor she cannot control, the only way $P_{\text{mal\_link}}$ will be close to 1 is, if she repeats the number of trials (x) to a polynomial number of times. Thus, the total amount of work in theoretical terms is poly-log(η_{web}). It is important to note that under these conditions, the number of susceptible users will be a linear function of the legitimate users and this number cannot be controlled by the miscreant. Further, the miscreant cannot aim at a specific number of susceptible users.

Similarly, if the miscreant is to aim for or target specific number of susceptible users, while assuming that there is certain activity from legitimate users, she is forced to increase the number of trials. In this particular scenario, the probability of users clicking on malicious links does not really factor in, since she is targeting a "specified number" of website users. This means that the miscreant has to match the posting activity of the legitimate users, i.e., $\Psi_{\text{website}} \approx \Phi_{\text{website}}$. Table 6.3

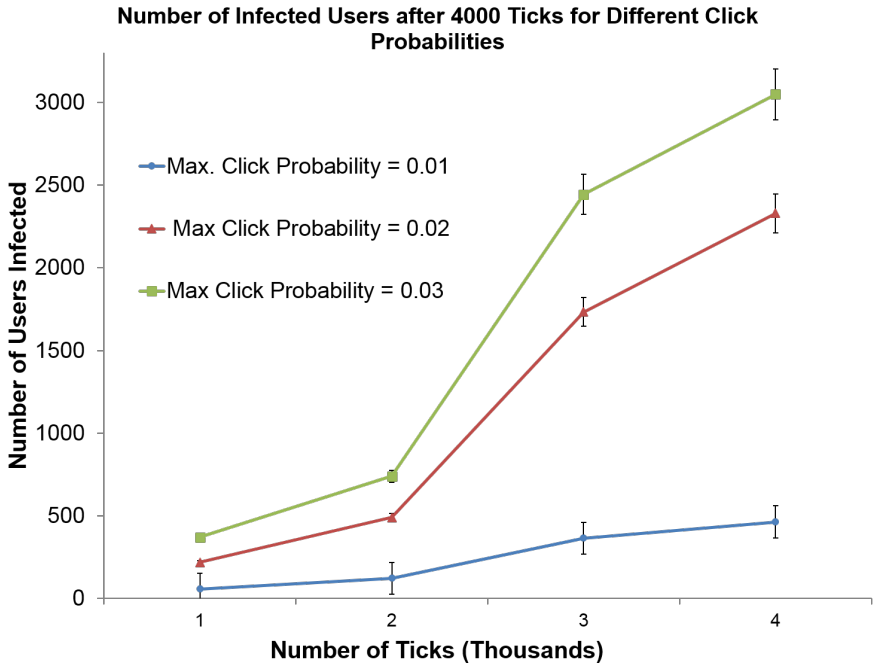

Figure 6.14: Number of infected users for different click probabilities

shows a summary of the resulting amount of work required from a miscreant, when she targets for a specific number of susceptibles while matching the legitimate users' posting activities. It is important to note that the legitimate users' posting activities will never go beyond some fraction of followers (η_{web}) which is a linear factor of η_{web}. It can be seen that if the relative user activity is less (say $\log(\eta_{web})$), a miscreant can still target a high number of susceptible users by repeating the attack an exponential number of times at a quasi-poly amount of work. Similarly, by targeting for a lower number of legitimate users she can save cost. Overall, the tables show that the work required by the miscreant while being large, is not impossible. Thus, probabilistically at least, the attacks on Twitter are definitely feasible for a determined miscreant.

6.8.4 Discussion

From the tables one can observe that the amount of work for a miscreant when targeting for a certain number of susceptibles is not prohibitively large under our attack scenarios. However, the cost analysis in the paper has not considered

Table 6.2: Cost Analysis Targeting

$\Phi(\eta_{web})$ (Assume)	$P_{\textbf{mal\_link}}$ (Aim)	$\Psi(\eta_{web})$ (Forced)	**No. of Trials** x	**No. of Susceptible users**	**Amount of Work (Cost)**
$\log(\eta_{web})$	≈ 1	$\log(\eta_{web})$	poly	$lin(\eta_{web})$	poly-$\log(\eta_{web})$
$lin(\eta_{web})$	≈ 1	$lin(\eta_{web})$	poly	$lin(\eta_{web})$	poly(η_{web})
$lin(\eta_{web})$	$\approx \frac{log(\eta_{web})}{lin(\eta_{web})}$	$\log(\eta_{web})$	poly	$\log(\eta_{web})$	poly-$\log(\eta_{web})$

certain factors or accounted for events that could affect the attack. The following paragraphs, presents some factors that have not been considered as well as the reasons behind the lack of real world experiments.

6.8.4.1 Factors not considered

One of the most important factors that has not been considered in this paper is *time*. While certain observations have been made regarding time, they are mostly limited to factors which can be quantified. This is due to the reason that it is not probabilistically or deterministically possible to model the duration of an attack or other durations such as the time taken to insert links, activity of other users, etc. Another aspect that has not been considered in this work is the diversity of devices involved. The emergence of smartphones and other hand-held devices has resulted in new methods of accessing and interacting with the Internet resources which might accelerate or decelerate malware propagation.

The factors that work against a miscreant also requires close attention. First, the attack proposed in this dissertation, to a large extent requires human expertise in activities such as choosing blogs, articles and semantically constructing the correct sentences. The effort that goes into this activity has been abstracted in this work. Second, mechanisms of banning the miscreant from making more posts by other users could adversely affect the attack. If such an event occurs in the middle of an attack, the attack could possibly get voided completely. The far-reaching repercussion however would also be the loss of the miscreant's account;

Table 6.3: Cost Analysis Targeting Number of Susceptible

No. of of Susceptible (Aiming)	$\Psi_{\textbf{website}}$ (Assumed)	**No. Trials** (Forced)	**Amount of Work**
exp(η_{web})	$log(\eta_{web})$	exp	Quasi-Poly ($2^{[log(\eta_{web})]^k}$)
exp(η_{web})	$lin(\eta_{web})$	exp	exp ($2^{(\eta_{web})^k}$)
poly(η_{web})	$log(\eta_{web})$	poly	poly-log(η_{web})
poly(η_{web})	$lin(\eta_{web})$	poly	poly(η_{web})

the creation and maintenance of which adds to the cost of an attack, which has been abstracted. Further, by tweaking the parameters of attacks and changing the posting strategies can also affect the attacks and the cost. A careful consideration and analysis are however required to understand the results of the variations and this part is outside the scope of the paper.

6.8.4.2 Lack of real-world experiments

Finally, the cost analysis in this paper is based on a probabilistic model which provides the best-case scenario for an attack and accounting for real world factors. Verifying this model requires real experiments and user study to ensure the validity of the model. However, it has to be noted that the experiments may present scenarios that are not captured by our model, since the factors such as activities on websites, clicking on links are dependent on individuals and their personality. Similarly, the results from these experiments may not be completely reproducible. These details are beyond the scope of this dissertation.

6.9 Summary

This chapter presented an attack model and analyses of Twitter as a malware propagation medium. First the impact of malware propagation on Twitter was analyzed to understand the feasibility of such an attack. The results of the simulation performed showed that such attacks would adversely impact the users of Twitter and would also be tough to mitigate. Second, attacks were conceptualized that leverage Twitter's inherent models, obfuscation of information by short-URLs and clickjacking methods that are common methodologies of web-based attacks in the real world. These attacks present strategies that model user behaviors and considered other avenues/entry-points of attacks. Also different factors that an attacker needs to consider to be successful were identified and were also validated using the NetLogo simulator that was built. The probabilistic model demonstrates that such attacks are feasible and that even with a low degree of connectivity an attacker can infect many users. Since the success of these attacks depends on the personal choices of people, it is difficult to obtain real world data regarding such attacks. This makes formulating effective mitigation techniques challenging, however the threat model results present a start in this regard.

7

Conclusion

In this chapter, the threat modeling techniques which have been presented so far are analyzed and their impact discussed. The chapter also highlights some of the open research issues and identifies some future work as well.

This dissertation has presented a paradigm shift towards threat modeling by incorporating the attacker's perspective into threat assessment which leads to a threat model that retains the benefit of being able to provide a realistic threat assessment and capability to be applied to any sort of networks, while being feasible in practical system usage. The concept of risk assessment on an entire system as used in other research while being a sound idea, suffers from deficiencies such as requiring a thorough understanding of the system and its component. The approach and techniques presented in this dissertation represent a primary basis for threat modeling, and by undertaking one aspect of the attack vector, viz., cost of information required for a successful attack, provides a fundamental step towards "proactive" security schemes (that aim at dissuading attacks against systems). However, the application of the threat model at various or all levels of a system, could overwhelm the defender with information, since there is an element of threat at every layer. The approach towards identifying the levels of abstraction where the model can be applied, tries to lessen this burden. Even though this dissertation does highlight the efficacy of the threat model by applying to various emerging networks, the model has yet to reach its culmination of becoming a completely practical threat modeling system. Serious issues such as insider attacks and wanton leakage of information still remain outside the purview

of the model. However, the design and approach of the threat model still makes it viable to be applied to various such issues as well as to ones that have not been foreseen yet.

After motivating and providing a general introduction to the topic of threat modeling and risk assessment in Chapter 1, this dissertation provided the needed background and a survey of the related work in Chapter 2. This chapter also showcased the deficiencies in current threat modeling approaches motivating the need for a new approach. It also provided some insights into state-of-the-art work that is currently done on some of the emerging networks and technologies. Chapter 3 presented the core idea of this dissertation by truly understanding the requirements and components of an attack and by analyzing the necessity of attack modeling. The paradigm shifting approach of involving the attacker's perspective towards a practical risk assessment is described in detail here. This approach leads to new techniques in attack modeling which are discussed later. Also discussed in this chapter is the abstraction levels the model can be applied. As mentioned above, the threat model described in this work does not address certain issues and needs to be extended before it can be a complete practical system.

Chapter 4 provides the first use of the techniques learnt in the creation of the threat assessment framework, by modeling attacks against smartphones. The chapter investigates the security of smartphones, in particular the SSH vulnerability of iDevices. This chapter also presented two novel attack scenarios involving smartphones and it was demonstrated that while smartphones do present an attractive target to launch DoS attacks against networks targeting the power consumption of smartphones is infeasible. However, given the battery capacity and form factor of a smartphone, the chapter showed how they provide an attractive option as an attack tool. While the work examined the effects of certain attacks, it still needs to be investigated how smartphones would react under other kinds of attacks.

In Chapter 5, the generic framework introduced in Chapter 3 was applied to MAC layer protocols and the risk of jamming attacks was assessed. The risk was assessed *low* by incorporating the attacker's perspective in the execution of such an attack. It was highlighted that the reasons for this assessment was due to

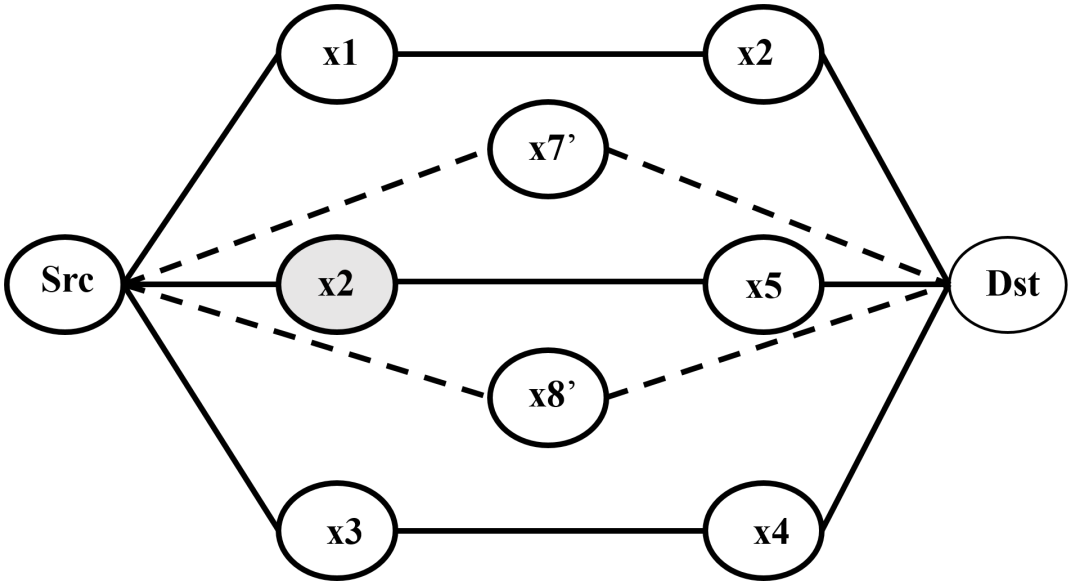

Figure 7.1: A sybil scenario

the fact that the attacker has better options and that for a successful jamming attack, the cost of gaining information is very high. This presents a thorough understanding of how one can model attacks that have no conventional defense mechanisms and it will be interesting to examine how the assessment can be made for attacks. One such class of attacks is the Sybil attacks [Douceur 2002]. In a sybil attack, a malicious node fabricates different identities in the form of multiple nodes. These fabricated nodes behave like normal nodes, but deceive and mislead other *physical* nodes. In the case of multipath routing, this attack can be especially harmful since, a sybil node can present better metrics albeit imaginary such that routes involving the fabricated nodes are always chosen Karlof and Wagner [2003]. Fig. 7.1 presents such a scenario, where node x2 is a sybil node and nodes x1, x2, x3, x4, x5 and x6 are physical nodes. The sybil node can fabricate the nodes x7' and x8', and advertise them with better metrics. In this case the multipath routing protocol would chose paths involving nodes x7' and x8' (represented by the dashed lines), rather than paths involving actual nodes (represented by bold lines). It is easy to see that with paths involving the fabricated nodes, the entire network gets partitioned. The risk assessment for Sybil Attacks would really help defenders in addressing mitigation techniques.

Chapter 6 presented a threat model for the online social network application Twitter. This chapter first investigated the potential of using Twitter for malware propagation, by analyzing both its structure and the means an attacker could leverage the inherent trust that is shared between the users of Twitter. The chapter then analyzed the impact of malware propagation by Twitter and it was shown that even with a low degree of connectivity an attacker can infect many users via Twitter. To accurately assess the threat to the application, attacks were conceptualized which leverage Twitter's inherent models, obfuscation of information by short-URLs and clickjacking methods that are common methodologies of web-based attacks in the real world. These attacks presented strategies that model user behaviors and considered other avenues/entry-points of attacks. The chapter then performed a thorough risk assessment by analyzing the cost and validating them via a Netlogo simulator and showed that the risk towards social networks is *very high*. The work in this chapter also showcased a lot of open research question such as the difficulty in modeling social networks, the lack of real world data and the inability of real world experiments providing a baseline risk assessment. These issues are the main reasons as to why sound mitigation techniques cannot be created for social networks, since the risk assessment from real-world experiments would vary. Tools such as those for modeling social networks and user behaviour as well as simulators for propagation of messages would be invaluable to the research community and would ultimately lead to practical mitigation techniques. As explained in Chapter 2, a behavioral game theory model along with some incentives could lead to some proactive malware mitigation techniques. It however, will depend on the maturity and capability of behavioral game theory constructs, to model complex human behavior accurately.

In conclusion, this dissertation has presented a set of techniques for attack modeling and threat modeling that is based on incorporating the attacker's perspective. The techniques range from modeling threats and attacks at three levels of abstraction that were identified, viz., Architecture, Protocol and Application. As discussed before, a number of issues remain to be addressed with respect to the creation of a complete practical threat modeling tool. Understanding the semantics of various attacks is necessary towards creating sound mitigation techniques. The techniques have been extensively modeled and validated in realistic

environments and have been shown to be feasible for practical implementation. Although scope for further improvements still remain, it is hoped that the techniques and approach that have been presented here advance the state-of-art, and provide practical utility to the system administrators and researchrs everywhere.

Relevant Publications

Ameya Sanzgiri and Shambhu Upadhyaya. Feasibility of attacks: What is possible in the real world–a framework for threat modeling. In *The 2011 International Conference on Security and Management,SAM*, 2011.

Steven Salerno, Ameya Sanzgiri, and Shambhu Upadhyaya. Exploration of attacks on current generation smartphones. *Procedia Computer Science*, 5(0): 546 – 553, 2011. ISSN 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2011.07.071. The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011).

Ameya Sanzgiri, Jacob Joyce, and Shambhu Upadhyaya. The early (tweet-ing) bird spreads the worm: An assessment of twitter for malware propagation. *Procedia Computer Science*, 10:705–712, 2012. The 23rd International Conference on Ambient Systems, Networks and Technologies (ANT-2012) / The 9th International Conference on Mobile Web Information Systems (MobiWIS 2012).

Ameya Sanzgiri, Andrew Hughes, and Shambhu Upadhyaya. Analysis of malware propagation in twitter. In *Reliable Distributed Systems (SRDS), 2013 IEEE 32nd International Symposium on*, pages 195–204, 2013. doi: 10.1109/SRDS.2013.28.

References

Bitly— Do More With Your Links. URL `https://bitly.com/`. 62

TinyURL.com - Shorten That Long URL in a Tiny URL. URL `http://tinyurl.com/`. 62

P. Ammann, J. Pamula, R. Ritchey, and J. Street. A host-based approach to network attack chaining analysis. In *Computer Security Applications Conference, 21st Annual*, pages 10–pp. IEEE, 2005. 21

Paul Ammann, Duminda Wijesekera, and Saket Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS 2002, pages 217–224, New York, NY, USA, 2002. ACM. 14

Apple. Apple ios software. 35

O. Arkin and F. Yarochkin. Xprobe2 - a 'fuzzy' approach to remote active operating system fingerprinting., 2002. http://www.sys-security.com. 32

Michel Barbeau, Jyanthi Hall, and Evangelos Kranakis. Detecting impersonation attacks in future wireless and mobile networks. In *In Proceedings of MADNES 2005 - Workshop on Secure Mobile Ad-hocNetworks and Sensors - Held in conjunction with ISC-2005*. SVLNCS, 2005. 13, 29

K. Beck. Analyzing tweets to identify malicious messages. In *2011 IEEE International Conference on Electro/Information Technology (EIT 2011), 15-17 May 2011*, 2011 IEEE International Conference on Electro/Information Technology (EIT 2011), page 5 pp., Piscataway, NJ, USA, 2011. IEEE. URL `http://dx.doi.org/10.1109/EIT.2011.5978594`. 23

B. Bell, E. Santos Jr, and S.M. Brown. 4.1 making adversary decision modeling tractable with intent inference and information fusion. *ADVERSARIAL INTENT INFERENCE FOR PREDICTIVE BATTLESPACE AWARENESS*, page 4, 2005. 21

Boldizsár Bencsáth, Gábor Pék, Levente Buttyán, and Márk Félegyházi. Duqu: Analysis, detection, and lessons learned. In *ACM European Workshop on System Security (EuroSec)*. ACM, 2012. 5

F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on Twitter. In *Proceedings of the Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, Washington, DC, USA, 2010. 23

Bettina Beurer-Zuellig and Miriam Meckel. Smartphones enabling mobile collaboration. *Hawaii International Conference on System Sciences*, 0:49, 2008. ISSN 1530-1605. doi: http://doi.ieeecomputersociety.org/10.1109/HICSS.2008.399. 21

Robert Beverly. A robust classifier for passive tcp/ip fingerprinting. In *PAM*, pages 158–167, 2004. 32

DEV TEAM BLOG. Dev team blog, 2011. 37

Linda Briesemeister, Patrick Lincoln, and Phillip Porras. Epidemic profiles and defense of scale-free networks. In *Proceedings of the 2003 ACM workshop on Rapid malcode*, WORM '03, pages 67–75, New York, NY, USA, 2003. ACM. ISBN 1-58113-785-0. URL `http://doi.acm.org/10.1145/948187.948200`. 23

M. Brownfield, Yatharth Gupta, and N. Davis. Wireless sensor network denial of sleep attack. In *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*, pages 356–364, 2005. 50

Juan Caballero, Chris Grier, Christian Kreibich, and Vern Paxson. Measuring Pay-per-Install: The Commoditization of Malware Distribution. In *Proceedings of the the 20th USENIX Security Symposium*, San Francisco, CA, August 2011. 59

J. B. D. Cabrera, L. Lewis, Qin Xinzhou, Lee Wenke, R. K. Prasanth, B. Ravichandran, and R. K. Mehra. Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study. In *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on*, pages 609–622, 2001. 40

Mario Cagalj, Saurabh Ganeriwal, and Jean pierre Hubaux. On selfish behavior in csma/ca networks. In *In Proc. of IEEE Infocom*, 2005. 24

Colin Camerer. Behavioral game theory: Experiments in strategic interaction. 24

Agnes Chan, Xin Liu, Guevara Noubir, and Bishal Thapa. Broadcast control channel jamming: Resilience and identification of traitors. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 2496 – 2500, June 2007. doi: 10.1109/ISIT.2007.4557594. 50

Senthilkumar G Cheetancheri. Modelling a computer worm defense system. Master's thesis, UNIVERSITY OF CALIFORNIA, DAVIS, 1998. 14

Shuo Chen, Zbigniew Kalbarczyk, Jun Xu, and Ravishankar K. Iyer. A data-driven finite state machine model for analyzing security vulnerabilities. In *In IEEE International Conference on Dependable Systems and Networks*, pages 605–614, 2003. 30

Yingying Chen, Wenyuan Xu, Wade Trappe, and YanYong Zhang. *Securing Emerging Wireless Systems: Lower-layer Approaches*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 0387884904, 9780387884905. 50, 51

C.Meadows. Applying formal methods to the analysis of a key management protocol. *Journal of Computer Security*, 1(1):5–36, 1992. 32

Jerald Dawkins. *Heuristics for scalable compound exposure analysis: a foundation for a comprehensive security risk assessment*. PhD thesis, Tulsa, OK, USA, 2005. AAI3163148. 14

P. De, Liu Yonghe, and S. K. Das. An epidemic theoretic framework for vulnerability analysis of broadcast protocols in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 8(3):413–425, 2009. ISSN 1536-1233. 23

MEMBERS OF CHRONIC DEV. Chronic dev blog, 2011. 37

Android Developers. Android and developement. 35

R. Di Pietro and N. V. Verde. Introducing epidemic models for data survivability in unattended wireless sensor networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1–6, 20-24 June 2011 2011. 23

Bryan Dixon and Shivakant Mishra. On rootkit and malware detection in smartphones. In *Proceedings of the 2010 International Conference on Dependable Systems and Networks Workshops (DSN-W)*, DSNW 10, pages 162–163, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-1-4244-7729-6. 22

Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, Fabian Kuhn, and Calvin Newport. The Wireless Synchronization Problem. In *Proceedings of the 28th Annual Symposium on Principles of Distributed Computing*, 2009. 31

John R. Douceur. The sybil attack. In *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers*, pages 251–260. Springer, 2002. 96

Qi Duan, Mohit Virendra, and Shambhu J. Upadhyaya. On the hardness of minimum cost blocking attacks on multi-path wireless routing protocols. In *ICC*, pages 4925–4930, 2007. 15

William Enck, Machigar Ongtang, and Patrick McDaniel. Understanding android security. *IEEE Security and Privacy*, 7(1):50–57, 2009. ISSN 1540-7993. 35

Shelby Evans, David Heinbuch, Elizabeth Kyule, John Piorkowski, and James Wallner. Risk-based systems security engineering: Stopping attacks with intention. *IEEE Security and Privacy*, 2(6):59–62, November 2004. ISSN 1540-7993. doi: 10.1109/MSP.2004.109. URL `http://dx.doi.org/10.1109/MSP.2004.109`. 21

M. R. Faghani and H. Saidi. Malware propagation in online social networks. In *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on*, pages 8–14, 13-14 Oct. 2009 2009. 23

Kevin Fall and Sally Floyd. Simulation-based comparisons of tahoe, reno and sack tcp. *SIGCOMM Comput. Commun. Rev.*, 26(3):5–21, 1996. ISSN 0146-4833. doi: http://doi.acm.org/10.1145/235160.235162. 31, 32

Abraham D. Flaxman, Alan M. Frieze, and Juan Vera. A Geometric Preferential Attachment Model of Networks. In *Algorithms and Models for the Web-Graph: Third International Workshop, WAW 2004*, pages 44–55. Springer, 2004. 84

Jay Freeman. Cydia, 2010. 37

Daniel Geer and John Harthorne. Penetration testing: A duet. In *ACSAC*, pages 185–198, 2002. 14

Leigh Goessl. Flame virus discovered, described as new 'super cyber-weapon'. WebPage, May 2012. URL `http://www.digitaljournal.com/article/325667`. 5

C. Griffin and R. Brooks. A note on the spread of worms in scale-free networks. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(1):198–202, 2006. ISSN 1083-4419. 23

Suvajit Gupta and Joel Winstead. Using attack graphs to design systems. *IEEE Security & Privacy*, 5(4):80–83, 2007. doi: 10.1109/MSP.2007.100. 14

Shawn Hernan, Scott Lambert, Tomasz Ostwald, and Adam Shostack. Uncover Security Design Flaws Using The STRIDE Approach, 2006. URL `http://msdn.microsoft.com/en-gb/magazine/cc163519.aspx`. 15

M. Howard, J. Pincus, and J.M. Wing. Measuring relative attack surfaces. *Computer Security in the 21st Century*, 2003. 6

M. Howard, J. Pincus, and J.M. Wing. *Computer Security in the 21st Century*. Springer, March 2005. 15

Verizon HTC. Htc mobile phones, 2011. 41

Qiang Huang, H. Kobayashi, and Bede Liu. Modeling of distributed denial of service attacks in wireless networks. In *Communications, Computers and signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on*, volume 1, pages 41–44 vol.1, 2003. 22, 49

ih8sn0w.com. ih8sn0w.com — jailbreak your ipod touches and iphones. 37

K. Ingols, R. Lippmann, and K. Piwowarski. Practical attack graph generation for network defense. In *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*, pages 121–130. IEEE, 2006. 14

S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*, pages 49 – 63, 2002. doi: 10.1109/CSFW.2002.1021806. 6, 14

Cong Jin, Xiaoyan Huang, and Songlin Jin. Propagation model of mobile phone virus based on efficiency of immunization. In *Proceedings of the 2008 International Conference on MultiMedia and Information Technology*, MMIT 2008, pages 500–502, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3556-2. 22

C. Karlof and D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 113 – 127, May 2003. doi: 10.1109/SNPA.2003.1203362. 96

D.L. Kewley and J. Lowry. Observations on the effects of defense in depth on adversary behavior in cyber warfare. 21

Peter B. Key and Derek McAuley. Differential qos and pricing in networks: Where flow control meets game theory. *IEE Proceedings - Software*, 146(1): 39–43, 1999. 24

M. H R Khouzani, S. Sarkar, and E. Altman. A dynamic game solution to malware attack. In *INFOCOM, 2011 Proceedings IEEE*, pages 2138–2146, 2011. doi: 10.1109/INFCOM.2011.5935025. 25

Pushmeet Kohli, Michael Kearns, Yoram Bachrach, Ralf Herbrich, David Stillwell, and Thore Graepel. Colonel blotto on facebook: The effect of social relations on strategic interaction. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, pages 141–150, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1228-8. doi: 10.1145/2380718.2380738. URL `http://doi.acm.org/10.1145/2380718.2380738`. 24

Sian Lun Lau and Klaus David. Movement recognition using the accelerometer in smartphones. In *Future Network and Mobile Summit 2010*, pages 1–9, Florence, Italy, June 16-18 2010. 21

D. Lee and K. Sabnani. Reverse-engineering of communication protocols. In *Network Protocols, 1993. Proceedings., 1993 International Conference on*, pages 208–216, 1993. 32

D. Lee, A. N. Netravali, K. K. Sabnani, B. Sugla, and A. John. Passive testing and applications to network management. In *ICNP '97: Proceedings of the 1997 International Conference on Network Protocols (ICNP '97)*, page 113, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-8061-X. 32

David Lee, Dongluo Chen, Ruibing Hao, Raymond E. Miller, Jianping Wu, and Xia Yin. A formal approach for passive testing of protocol data portions. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 122–131, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1856-7. 32

Kristina Lerman and Rumi Ghosh. Information contagion: an empirical study of the spread of news on digg and twitter social networks. In *Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM)*, 2010. 23

Athanasios Loukas, Dimitrios Damopoulos, Sofia Menesidou, Maria Skarkala, Georgios Kambourakis, and Stefanos Gritzalis. Milc: A secure and privacy-preserving mobile instant locator with chatting. *Information Systems Frontiers*, pages 1–17, 2010. ISSN 1387-3326. 22

Gavin Lowe and Bill Roscoe. Using csp to detect errors in the tmn protocol. *IEEE Trans. Softw. Eng.*, 23(10):659–669, 1997. ISSN 0098-5589. 32

J. Lowry, R. Valdez, and B. Wood. Adversary modeling to develop forensic observables. 2011. 21

A.B. MacKenzie and S.B. Wicker. Game theory and the design of self-configuring, adaptive wireless networks. *Communications Magazine, IEEE*, 39(11):126–131, 2001. ISSN 0163-6804. doi: 10.1109/35.965370. 24

A.B. MacKenzie and S.B. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1583–1590 vol.3, 2003. doi: 10.1109/INFCOM.2003.1209181. 24

P Manadhata and J Wing. An attack surface metric. *Software Engineering, IEEE Transactions on*, PP(99):1, 2010. ISSN 0098-5589. doi: 10.1109/TSE.2010.60. 6, 15

Mohammad Mannan and Paul C. van Oorschot. On instant messaging worms, analysis and countermeasures. In *Proceedings of the 2005 ACM workshop on Rapid malcode*, WORM '05, pages 2–11, New York, NY, USA, 2005. ACM. ISBN 1-59593-229-1. URL `http://doi.acm.org/10.1145/1103626.1103629`. 23, 59

S. Massoud Amin and B.F. Wollenberg. Toward a smart grid: power delivery for the 21st century. *Power and Energy Magazine, IEEE*, 3(5):34–41, 2005. 12

J.P. McDermott. Attack net penetration testing. In *Proceedings of the 2000 workshop on New security paradigms*, pages 15–21. ACM, 2001. 18

Catherine Meadows. Formal methods for cryptographic protocol analysis: emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, 21 (1):44–54, January 2003. 32

G. Cohen B. Meiseles and E. Reshef. System and method for risk detection and analysis in a computer network,, October 2005. 14

Microsoft. Microsoft windows mobile 7. 35

D.P. Mirembe and M. Muyeba. Threat modeling revisited: improving expressiveness of attack. In *Computer Modeling and Simulation, 2008. EMS'08. Second UKSIM European Symposium on*, pages 93–98. IEEE, 2008a. 19

Drake Patrick Mirembe and Maybin Muyeba. Threat modeling revisited: Improving expressiveness of attack. *Computer Modeling and Simulation, UKSIM European Symposium on*, 0:93–98, 2008b. doi: http://doi.ieeecomputersociety.org/10.1109/EMS.2008.83. 15

R.A. Moore, DL Kewley, R.C. Parks, and LS Tinnel. The information battlespace preparation experiment. In *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, volume 1, pages 352–366. IEEE, 2001. 21

I. Morikawa and Y. Yamaoka. Threat tree templates to ease difficulties in threat modeling. In *Network-Based Information Systems (NBiS), 2011 14th International Conference on*, pages 673–678. IEEE, 2011. 20

Research In Motion. Research in motion- blackberry. 35

Collin Mulliner. Vulnerability analysis and attacks on nfc-enabled mobile phones. *2009 International Conference on Availability Reliability and Security*, 32(1):695–700, 2009. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5066549`. 22

T. Nakashima and S. Oshima. A detective method for syn flood attacks. In *Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference on*, volume 1, pages 48–51. 39

Dalia Nashat, Xiaohong Jiang, and Susumu Horiguchi. Detecting syn flooding agents under any type of ip spoofing. In *Proceedings of the 2008 IEEE International Conference on e-Business Engineering*, ICEBE '08, pages 499–505,

Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3395-7. doi: 10.1109/ICEBE.2008.18. URL http://dx.doi.org/10.1109/ICEBE.2008.18. 39

Liam O. Murchu Nicolas Falliere and Eric Chien. W32.Stuxnet Dossier, 2011. 5

Peng Ning and Wenliang Du. Journal of computer security, January 2007. 54

NMAP. Nmap- free security scanner for network exploration and security audits, 2010. 42

Guevara Noubir and Guolong Lin. Low-power DoS attacks in data wireless LANs and countermeasures. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7:2930, July 2003. ISSN 1559-1662. doi: http://doi.acm.org/10.1145/961268.961277. URL http://doi.acm.org/10.1145/961268.961277. 23, 50

A.J. O'Donnell. When malware attacks (anything but windows). *Security Privacy, IEEE*, 6(3):68–70, 2008. ISSN 1540-7993. doi: 10.1109/MSP.2008.78. 25

X. Ou, S. Govindavajhala, and A.W. Appel. Mulval: A logic-based network security analyzer. In *Proceedings of the 14th conference on USENIX Security Symposium-Volume 14*, pages 8–8. USENIX Association, 2005. 14

J. Padhye and S. Floyd. On inferring tcp behavior. In *In the proceeding of SIGCOMM*, pages 287–298, 2001. 32

Vaibhav Ranchhoddas Pandya and Mark Stamp. iphone security analysis. *J. Information Security*, 1(2):74–87, 2010. 37

Cyrus Peikari and Seth Fogie. *Maximum Wireless Security*. Sams, Indianapolis, IN, USA, 2002. ISBN 0672324881. 28

Sara Peters. *2010 CSI/FBI Computer Crime and Security Survey*. Computer Security Institute, December 2009. 52

Sara Peters. Csi computer crime and security survey 2010 – 2011, 2011. URL http://gocsi.com/survey. 58

R. A. Poisel. *Modern Communications Jamming Principles and Techniques*. Artech House Publishers, 2006. 51

John G. Proakis and Dimitris K. Manolakis. *Digital Signal Processing (4th Edition)*. Prentice Hall, March 2006. ISBN 0131873741. 54

T. Rao and S. Nagpal. Real-time geo influence in social networks. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 1, pages 246–250, 8-10 April 2011 2011a. 66

T. Rao and S. Nagpal. Real-time geo influence in social networks. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 1, pages 246–250, 8-10 April 2011 2011b. 66

M. Raya and J.P. Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 11–21. ACM, 2005. 12

D. Raymond, R. Marchany, M. Brownfield, and S. Midkiff. Effects of denial of sleep attacks on wireless sensor network MAC protocols. In *Information Assurance Workshop, 2006 IEEE*, pages 297–304, 2006. 50

D.R. Raymond and S.F. Midkiff. Denial-of-Service in wireless sensor networks: Attacks and defenses. *Pervasive Computing, IEEE*, 7(1):74–81, 2008. ISSN 1536-1268. 50

R.W. Ritchey and P. Ammann. Using model checking to analyze network vulnerabilities. In *Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on*, pages 156 –165, 2000. doi: 10.1109/SECPRI.2000.848453. 14

Bruce Schenier, 2008. URL `http://www.schneier.com/crypto-gram-0005.html`. 7

Bruce Schneier. *Secrets and lies - digital security in a networked world: with new information about post-9/11 security*. Wiley, 2004. ISBN 978-0-471-45380-2. 16, 18

S. Shah. An introduction to http fingerprinting, 2004. http://net-square.com/httprint/httprint paper.html. 32

O. Sheyner and J. Wing. Tools for generating and analyzing attack graphs. In *Formal methods for components and objects*, pages 344–371. Springer, 2004. 21

Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, and Jeannette M. Wing. Automated generation and analysis of attack graphs. *Security and Privacy, IEEE Symposium on*, 0:273, 2002. ISSN 1540-7993. doi: http://doi.ieeecomputersociety.org/10.1109/SECPRI.2002.1004377. 14, 30

Woohyun Shim, L. Allodi, and F. Massacci. Crime pays if you are just an average hacker. In *Cyber Security (CyberSecurity), 2012 International Conference on*, pages 62–68, 2012. doi: 10.1109/CyberSecurity.2012.15. 25

Cheng Shin-Ming, Ao Weng Chon, Chen Pin-Yu, and Chen Kwang-Cheng. On modeling malware propagation in generalized social networks. *Communications Letters, IEEE*, 15(1):25–27, 2011. ISSN 1089-7798. 23, 59

Adam Shostack, 2011. URL `http://technet.microsoft.com/en-us/security/hh778966.aspx`. 7

G. Shu. *Formal Methods And Tools For Testing Communication Protocol System Security*. PhD thesis, Ohio State University, 2008. 32

Guoqiang Shu and David Lee. Network protocol system fingerprinting – a formal approach. In *Proceedings of IEEE Infocom*, 2006. 32

Daniel Siegel. On the new threats of social engineering exploiting social networks. Bachelor's thesis, Technische Universität München, August 2009. 62

A. Singh and A. Lakhotia. Game-theoretic design of an information exchange model for detecting packed malware. In *Malicious and Unwanted Software (MALWARE), 2011 6th International Conference on*, pages 1–7, 2011. doi: 10.1109/MALWARE.2011.6112319. 25

Brian Skyrms and Robin Pemantle. A dynamic model of social network formation. In *Adaptive Networks*, pages 231–251. Springer, 2009. 24

Jan Steffan and Markus Schumacher. Collaborative attack modeling. In *Proceedings of the 2002 ACM symposium on Applied computing*, SAC '02, pages 253–259, New York, NY, USA, 2002. ACM. ISBN 1-58113-445-2. doi: 10.1145/508791.508843. URL `http://doi.acm.org/10.1145/508791.508843`. 16

J. Steven. Threat modeling - perhaps it's time. *Security Privacy, IEEE*, 8(3):83 –86, may-june 2010. ISSN 1540-7993. doi: 10.1109/MSP.2010.110. 1, 4, 7

G Stoneburner, A Goguen, and A Feringa. Risk management guide for information technology systems. *Nist Special Publication*, 19(800-30):58, 2002. 21

Yogesh Prem Swami and Hannes Tschofenig. Protecting mobile devices from tcp flooding attacks. In *Proceedings of first ACM/IEEE international workshop on Mobility in the evolving internet architecture*, MobiArch '06, pages 63–68, New York, NY, USA, 2006. ACM. ISBN 1-59593-566-5. 22

TeslaCoil. Quicksshd — teslacoil software. 38

Seth Tisue and Uri Wilensky. NetLogo: A Simple Environment for Modeling Complexity. In *in International Conference on Complex Systems*, pages 16–21, 2004. 82

unrevoked. Unrevoked set your phone free. 38

Sven Van Segbroeck, Francisco C Santos, Tom Lenaerts, and Jorge M Pacheco. Reacting differently to adverse ties promotes cooperation in social networks. *Physical review letters*, 102(5):058105, 2009. 25

David Watson, Matthew Smart, G. Robert Malan, and Farnam Jahanian. Protocol scrubbing: network security through transparent flow modification. *IEEE/ACM Trans. Netw.*, 12(2):261–273, 2004. ISSN 1063-6692. doi: http://dx.doi.org/10.1109/TNET.2003.822645. 32

Dongho Won and Seungjoo Kim, editors. *Information Security and Cryptology - ICISC 2005, 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers*, volume 3935 of *Lecture Notes in Computer Science*, 2006. Springer. ISBN 3-540-33354-1. 14, 17, 18

A.D. Wood and J.A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002. ISSN 0018-9162. 23, 50, 51

A.D. Wood, J.A. Stankovic, and S.H. Son. JAM: a jammed-area mapping service for sensor networks. In *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*, pages 286–297, 2003. 23, 50

Yongkang Xiao, Xiuming Shan, and Yong Ren. Game theory models for ieee 802.11 dcf in wireless ad hoc networks. *Communications Magazine, IEEE*, 43 (3):S22–S26, 2005. 24

Wei Xu, Fangfang Zhang, and Sencun Zhu. Toward worm detection in online social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 11–20, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0133-6. 23

Wenyuan Xu, Timothy Wood, Wade Trappe, and Yanyong Zhang. Channel surfing and spatial retreats: defenses against wireless denial of service. In *Workshop on Wireless Security'04*, pages 80–89, 2004. 23, 50

Wenyuan Xu, Ke Ma, W. Trappe, and Yanyong Zhang. Jamming sensor networks: attack and defense strategies. *Network, IEEE*, 20(3):41–47, 2006. ISSN 0890-8044. 50, 51

Guanhua Yan, Guanling Chen, Stephan Eidenbenz, and Nan Li. Malware propagation in online social networks: nature, dynamics, and defense implications. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 196–206, Hong Kong, China, 2011. ACM. ISBN 978-1-4503-0564-8. URL `http://dx.doi.org/10.1145/1966913.1966939`. 23

F. Yarochkin. Remote os detection via tcp/ip stack fingerprinting., 1998. http://www.insecure.org. 32

Liqiang Zhao, Jie Zhang, Kun Yang, and Hailin Zhang. Using incompletely cooperative game theory in mobile ad hoc networks. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 3401–3406. IEEE, 2007. 24

Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125–138, New York, NY, USA, 2004. ACM. ISBN 1-58113-793-1. doi: http://doi.acm.org/10.1145/990064.990081. 31

C. C. Zou, D. Towsley, and Gong Weibo. Email worm modeling and defense. In *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pages 409–414, 11-13 Oct. 2004 2004. ISBN 1095-2055. 23