# I'm OK, You're OK, the System's OK: Normative Security for Systems

Olgierd Pieczul[1,2]
olgierdp@ie.ibm.com

Simon N. Foley[2]
s.foley@cs.ucc.ie

Vivien M. Rooney[3]
vivrooney@gmail.com

[1] Ireland Software Lab, IBM Software Group, Dublin, Ireland
[2] Department of Computer Science, University College Cork, Ireland
[3] Insight Centre, University College Cork, Ireland

## ABSTRACT

The normative security paradigm seeks to view a system as a society in which security is achieved by a combination of legislative provisions and normative behaviors. Drawing solely on legislative provisions is insufficient to achieve a just and orderly society. Similarly, security paradigms that focus solely on security policies and controls are insufficient. We argue that systems have analogous normative behaviors—behavioral norms—that are learnt from system logs. Using this analogy we explore how current theories about social norms in society can provide insight into using normative behavior in systems to help achieve security.

## 1. INTRODUCTION

The increasing scale and complexity of modern computer systems means that the provision of effective security is challenging, as well as being prohibitively expensive. Consequently, security tends to regulate those activities perceived to be critical, with the assumption that other unregulated activities, whether known or unknown, are not of significance. An added complication with security regimes that are overly strict, is that such unregulated activities can become the means of getting things done in the system.

However, the difficulty is that these side-activities often lead to the compromise of security in a system. While security controls may provide monitoring and enforcement of the critical activities related to the security policy, little may be known about the nature of the other activities.

Our position is that the 'security' of a system is based not only on the regulation of what is perceived to be its security critical activities, but also on the orderliness of its unregulated activities. We characterize this orderliness in terms of *behavioral norms* [31], corresponding to repeating patterns of behavior that emerge in the system over time. Previous research [31,32] considered how these behavioral norms, representing potentially unknown side-activities, can be revealed by mining detailed system logs. The assumption is that, absent other information, adherence to past normative behavior can be taken as some indication of continuing orderliness. However, we note that these behavioral norms can be used to gauge the order or disorder in a system and, therefore, adherence to past normative behavior may also indicate a continuation of disorderliness

In the seminal work of Forrest at al. [16,17], immunology concepts from the field of Biomedical Sciences provided a novel way to think about systems security and led to the formulation of anomaly detection as a self/non-self decision. There are many other examples whereby analogical thinking [39] has been successfully used to provide new insight and solutions to problems in unrelated domains. We follow this same strategy of using the analogy of social norms to provide a different perspective on systems security.

This paper considers the foundations for a new security paradigm based on the identification and monitoring of normative behavior in systems. We use the term *system* in its most general sense, including user activities, business processes, and computer-based components. The paradigm treats a system as similar to a society in which security and *orderliness* is sought. In society, while much of daily life is governed by *social norms* neither formally codified, nor generally enforced by institutions of the state, such norms are, nevertheless, important in maintaining social order. We argue that just as security and order in society are not maintained by the rule of law alone, meaningful system security—and order—cannot be achieved by reliance solely on traditional security paradigms that seek to prescribe regulation of activity. Therefore, just as society's social norms are part of the maintenance of social order, beyond the specifics of legislation; behavioral norms can represent conventions that enable the orderly operation of a system, beyond its security policy. Note that these conventions for orderly system operation are not limited to the behavioral norms of individuals, but extend across all the components of the system.

The paper is organized as follows. Section 2 considers some of the problems arising from traditional system security paradigms, and draws comparisons with challenges in society that arise from legislative provisions. In practice, achieving security and order in society extends beyond policies enacted as legal provisions. Section 3 argues that behavioral norms should be treated as the system equivalent of social norms in society. Building on this analogy, Section 4 considers how theories about social norms can shed light on behavioral norms in systems, and thereby provide a founda-


*NSPW '14* September 15 - 18 2014, Victoria, BC, Canada
ACM 978-1-4503-3062-6/14/09 ... $15.00
http://dx.doi.org/10.1145/2683467.2683476.

tion for the development of a new paradigm of orderliness. Section 5 concludes the paper.

## 2. SECURE SYSTEMS, SECURE SOCIETIES

Threats, policies, and enforcement are the focus in system security. Similarly, security in society focuses on threats, laws and enforcement. This section draws on some of the similarities between security in society and security in systems, and observes that in practice, providing true security extends beyond policies and laws. True security draws on the normative behavior that emerges in society.

### 2.1 The Reference Monitor as a police state

A traditional approach to security follows a *reference monitor style paradigm*: the development of a security mechanism enforcing a security policy, with the objective of keeping the system in a secure state [5]. The security policy is comprised of rules that define what is, and what is not, permitted. This is comparable to a legal system in society whereby laws—explicit rules that define what is, and what is not, legal—are enforced by state agencies, such as police. At the extreme, the paradigm's closed world rules and mechanisms can be compared to a police state, whereby legal provisions are the only reference point, and these provisions are enforced by institutions of the state. However, the maintenance of order in society is ordinarily based on a wide ranging body of rules, laws, practices and customs. Much of these overlap, and are underpinned by social norms. It is often the case that the process of maintaining order is flexible and adaptable, as compared with the inherent inflexibility of the reference monitor paradigm.

A significant challenge associated with the reference monitor paradigm is to understand the security threats, and decide upon an appropriate security policy and mechanism. Even after much study there can still be disagreement and misunderstanding on the meaning of security. For example, the many definitions for secure information flow [13, 21, 26], or how misinterpretation of a threat model can lead to invalid conclusions about a security protocol [19]. Regardless of the technical challenges, inadequate security requirements elicitation can also lead to a mismatch between user expectation and enforcement. For example, while a requirements engineer might consider access control lists acceptable, non-technical users can have quite different ideas about security policies [15].

Attempting to address all possible threats—assuming they are known—can lead to overly complex and contradictory policies. Trade-offs must be made, for example, between the threat associated with storing personal data, and regulatory audit requirements. In practice, security adapts over time, reflecting evolving needs and understanding.

Similar challenges exist in the law. Where common law systems operate, case law is developed incrementally. Decisions in particular court cases reflect particular circumstances, and thereby expand the body of decisions forming the law. Applying the doctrine of Stare Decisis, new case law adds to the body of existing legal precedent. This process can take account of new situations as society evolves.

In systems, even when security requirements are understood, they must be encoded accurately in terms of enforceable policies. A policy language is used to prescribe secure and non-secure states. This should be expressive, and not give rise to ambiguity or inconsistency. However, in practice and regardless of language precision, errors in policy articulation are made. For example, [43] describes user misunderstanding of security policy settings in PGP email.

The inadequacy of legislation can also be revealed as a rule is interpreted under very particular circumstances. Legal Formalism [9] regards law as a closed system, meaning that resolution of legal questions references legal concepts and the facts of a case. Formality is claimed to preserve objectivity and impartiality, and as such underpin the legitimacy of law. An example of the inadequacy of this approach is given by Hutchinson [23] in the context of a dispute that arose following a swimming competition at the University of Toronto. The person placed second in the race sought recourse to the rules of the competition to challenge the winner of the race. Following lengthy debate and poring over the rules, the challenger was declared the winner. The rules stated that the winner was the person who first touched the wall of the pool with both hands. The winning swimmer had only one arm.

In practice, the range of problems considered by the reference monitor paradigm can be quite narrow. For example, a security kernel controls low-level system activities, while attribute-based access control provide application-level policies. Separating concerns in this way may simplify policy elicitation and articulation, however it can encourage an excess of security controls. Furthermore, interoperation between heterogeneous policies can lead to anomalies that cannot be easily addressed by conflict resolution mechanisms. For example, a study [20] on network firewall configuration shows that almost 30 percent of expert system administrators made configuration mistakes that led to serious policy conflicts.

Adherence to rule-based legal systems can also result in undesirable or unintended outcomes, as illustrated by the swimming competition case. The positivist argument underpinning such decisions is that judges are obliged to apply an enacted law, even if it is unjust, in order to preserve the legitimacy of the legal system. However, there are circumstances, such as the wider social context, when a law may be sidelined. For example, an Australian judge dismissed charges against four women who had defaced an advertising billboard. In this case, the billboard depicted a woman, wearing Berlei underwear, being sawn in half, with the caption 'You'll always feel good in Berlei'. The graffiti added was 'Even if you're mutilated'. On the basis of violence against women being endemic, the judge dismissed the charges. In this instance, the judge cited the ubiquity of representations of violence against women to sideline positivist law [27]. This illustrates the inherent flexibility of how order is maintained in an evolving society. In the same manner, system security is also regarded as a process [35]; it is expected that security policies and controls change as new threats and requirements are identified. In changing circumstances or emergency situations, a lack of flexibility may lead taking alternative paths to realize business goals, that bypass security policy [24].

Under the reference monitor paradigm, security mechanisms ensure that the system remains in a secure state by upholding their respective security policies. This is a somewhat stronger condition than enforcement of laws where the state attempts to create conditions such that society will follow the law. Thus, we make the loose connection between *active* enforcement of security in systems and the *construc-*

*tive* enforcement of laws in society. Both security policies and laws can fail if their enforcement mechanisms are missing, prove to be ineffective, or have flaws in their implementation. There are many examples of how programming flaws can lead to security vulnerabilities. Similarly, in applying the US exclusionary rule of evidence, there are many examples of how flaws in implementing due process can render evidence inadmissible in court.

## 2.2 Security Risk Management as a bureaucratic state

While the traditional Reference Monitor paradigm includes security controls and policies that defend against known design-time threats, it does not deal with changes in threats, vulnerabilities and/or failures in controls. For example, perhaps ACLs were previously considered adequate to defend against insider-trading, however, subsequent identification of a Trojan horse threat requires deployment of mandatory access controls. In practice, security is a process [35] and the objective of a *Security Risk Management* paradigm is to treat security as just another risk that needs to be managed alongside other risks to business objectives. This Internal Controls style approach of *achieving reasonable assurance regarding the achievement of objectives* [1,14,38] follows an OODA-style loop [8]: firstly, identifying security risks (threats), secondly, selecting and deploying security controls that mitigate the risks, and finally, measuring the efficacy of the controls at mitigating risk. Blakely et. al. [7] argue that this type of paradigm should be applied in a manner similar to its application in the medical profession. Thus, security risk management would follow scientific method and be carried out by trained and licensed professionals, with an obligation to operate under a code of ethics.

Intuitively, taking this approach can be regarded as a scaling-up of the Reference Monitor paradigm. This is a practical strategy for managing a great number and variety of 'reference monitors', and their threats and failures, across the enterprise. We use this analogy loosely, to characterize the current trend of compliance-driven security management, whereby catalogues of standards and best-practices help guide identification and judgement of a wide range of threats, controls and their efficacy measurement [2, 29, 41].

The security risk management paradigm has it attractions; unlike the reference monitor paradigm, it does not treat security as a binary notion, and enables management of systems that are secure within some acceptable degree of risk. While compliance catalogues, standards and best practices may help a typical administrator to elicit, articulate and comprehend security risks, the extensive nature of these catalogues encourages a focus on checkbox-style security compliance, rather than security outcomes. At the extreme, approaches such as the Security Content Automation Protocol (SCAP) family of standards [41] champion catalogues with a great amount of detail, leading to challenges in comprehension. For example, the scope for inconsistencies within and between OVAL, CPE, CVE and CCE repositories in SCAP are considered in [12].

Overly regulated systems become unusable with the result that security controls are routinely bypassed in order to meet business goals. Users look for ways around the security controls as a means of doing the 'right thing and not the corporate process'. For example, administrators open additional, insecure channels to connect system components, developers disable or weaken security controls such as SSL certificate verification to simplify application testing. Overly rigid password policies may cause users to write their passwords down [4], and invasive access controls may result in them using administrator's accounts for their daily work.

This security paralysis is also evident in over-regulated societies where an unwieldy system ceases to function. A good example was the Court of Chancery in England, satirized by Charles Dickens in Bleak House. The labyrinth that was the Court of Chancery became inefficient for a variety of reasons, both procedural and owing to the volume of cases with which it dealt. This process sometimes took many years, while the assets under dispute were exhausted in the process.

Highly regulated environments may be, incorrectly, perceived as secure and justified by a 'theatre' of security controls. For example, systems may contain strong cryptographic mechanisms but fail to properly protect the keys. If security mechanisms are not usable, their users may apply them incorrectly, still believing that security goals are achieved [43]. Over-regulation may stop thinking-through, and standard good practices may be overlooked. Network administrators may use a feature-rich firewall as an excuse not to worry about local network topology; users may believe that anti-virus software will keep their PC secure and run software from untrusted locations.

This security theater also occurs in legal systems to varying degrees. For example, there may be laws in place to protect your rights. However, using these de jure systems may be prohibitively expensive and/or take place over a lengthy time frame. Justice delayed is justice denied. This leads to the de facto situation where no means of seeking redress is available to an injured party.

Earlier, we noted how common law systems can adapt incrementally to changing circumstances through the development of case law. The ability to respond to changing mores in society was illustrated by a judge relying on a growing critique of violence against women in order to ignore a positive law. Similarly, changes in how society perceives behaviour, that is, changing norms, can result in laws falling into disuse, and ultimately to them being amended. An example of this is the decriminalization of homosexuality in Ireland. Prosecutions under the relevant statute had become non-existent, however, the lengthy process of removing this provision from the statute books reflected changes in public opinion. Part of this change resulted from activists arguing their case before the legal system, failing, and consistently reiterating their arguments in different fora, from the Supreme Court in Ireland, to the European Court of Justice. This illustrates how social norms can change, and how this can, in turn, change formal legal provisions governing private conduct.

However, in the aftermath of controversial legal cases, the pressure of public opinion can produce questionable responses from a legislature. Social disquiet, associated with what is perceived as the failure of the legal system, can result in the introduction of statutory provisions aimed at addressing public concerns. Enacting statutes in such circumstances may satisfy public opinion, however, it may not provide an optimal framework for administering justice, or may even compound an existing difficulty. An example of this occurred in the aftermath of the acquittal of John Hinkley on a charge of attempting to assassinate President Regan. Having successfully raised a defense of insanity at trial, the

Insanity Defense Reform Act 1984 was subsequently passed. This Act restricted insanity as a defense, thereby addressing public concerns that it was being manipulated in criminal proceedings. It is questionable whether this reform of the criminal justice system addressed the needs of defendants suffering from mental illness. It is worth noting that this change in law does not affect only the few defendants who might choose to plead the defense. Rather, this alteration in legal statute reflects the concept of the mind in legal terms, and as we see how law, the legitimate and powerful arm of societal sanction, deals with the concept of 'insanity'. This concept feeds into our collective psyche as the law tells us what 'insanity' is, and therefore is part of evolving societal norms. Hence what may appear to have limited application or impact, in fact, has an effect in the wider sphere.

There are similar examples where well-intended changes to system (security) have led to unanticipated security vulnerabilities. For example, when the Debian team implemented a change in OpenSSL that was intended to improve memory management practices, a lack of understanding of the mechanism resulted in a compromise of the random number generation process leading to serious security vulnerability [3].

# 3. NORMS AND SOCIAL ORDER

Even if one attempts to manage enterprise security as if it were a police/bureaucratic state, in practice, the cost of gaining a complete understanding of all the components and operations of a large system is likely to be prohibitive. As a consequence, security tends to focus on those activities perceived to be critical, with an assumption that the other unregulated activities, known or unknown, are not significant. However, often it is these side-activities that can lead to a security compromise of the system. While security controls provide monitoring and enforcement of the critical activities related to the security policy, effectively, little is known about the nature of the other activities.

Our thesis is that, given the inadequacies of the conventional security paradigms, the security of the system rests heavily on whether these side-activities are *normal*.

The perception of security in society is gained, not just from legislation, but also from the presence of informal *social norms*. Studies, such as [11] show that informal norms may be efficient and cost-effective alternatives to legislation. For example, people queuing to an ATM often keep a distance from the person operating the machine. Having a private space available, the ATM user can feel more comfortable when entering the PIN and collecting cash. If the accepted distance is not kept, the ATM user may be alerted to a potentially risky situation, and no longer feel comfortable. It is the *combination* of the ATM security mechanism and the social norm around the unregulated side-activity of queuing, that contribute to the user feeling comfortable and therefore secure. This also reflects an often overlooked part of security which is that users are part of the system, exercising their own judgment and following informal and unexpected processes [44]. Note that social norms can also underpin behavior that is at odds with security policies [33]. Of course, a challenge with social norms is that they can be difficult to recognize and understand. In prescribing the security mechanisms for the first ATM machine, one wonders whether the designers considered the social norms of queue formation.

## 3.1 Social Norms

Social Psychology's focus is on how the thoughts, feelings and behaviors of individuals are influenced by the actual, imagined or implied presence of others [22]. One of the main components of Social Psychology is how we understand and participate in the social world. Attitudinal and behavioral uniformities form the basis for how people achieve that. We learn from others' behavior what is considered to be normal. Norms can be explicit rules, or implicit, integrated into everyday life. Norms have two components, the descriptive and the prescriptive. Thus via norms group members can learn what is, the descriptive, and what ought to be, the prescriptive. The function of norms can, as noted above, be understood as having two parts. Firstly, norms function such that we know that we understand the world correctly. Secondly, by participating in the norms, we know that we belong in the world. Norms function, therefore, to assure us of a valued common reality. We participate in this reality, and are accepted and gain approval. Normative discontinuities provide the contours of different social groups. Thus, as norms differ across groups, those differences delineate the groups. A society that is functioning well has a reasonable number of well understood social norms. To believe in a norm, and act according to its tenets, there are a number of requirements. The requirements are that the norm must be believed to be: (a) correct, (b) appropriate, (c) valid, and (d) socially desirable.

Norms provide a range of behavior that is acceptable in a certain context. As noted above, the function of the norm is that we understand the world. Thus, using the norm as a guide reduces uncertainty, and enhances confidence that a choice made is the correct course of action [33]. Applying this to the context of making decisions about choosing a system password, drawing attention to a norm within a group can be a useful tool [45]. Similarly, information about the privacy setting choices made by others in our group can provide valuable information, alerting us to the choices of those with whom we associate [25]. In a situation of uncertainty, informational influence provides evidence of reality. This confirms that we understand the situation correctly, for example, the necessity to pay attention to privacy settings, or to create a more secure password. The behavior of others is a frame of reference, and the middle position perceived as correct. Social norms emerge to guide behavior under conditions of uncertainty.

## 3.2 Behavioral Norms

*Behavioral norms* [31] represent patterns of behavior that can be discovered from event traces/logs. Norms provide abstract approximations of system behavior that is exhibited in logs.

An approach for inferring behavioral models from system logs was proposed in the seminal work of Forrest at al. [17]. System behavior is modeled in terms of a set of *n-grams* which represent short-range correlations between system call operations present in the system log. As the system executes, its operations are compared against this model of 'normal' and, if the sequence does not match known n-grams, it may be considered anomalous. This approach—modeling a system as a single amalgamation of all behavior—has limitations, as illustrated by the following example.

Figure 1 depicts a part of the log that could have been generated by a web-based order processing system. Each

```
time      user   role    method  path1/path2
-----------------------------------------------
10:44:40  alice  client  "PUT   /order/4c4712"   ◁
10:48:09  rob    admin   "GET   /userlist"
10:49:15  frank  client  "PUT   /order/1d261e"   ◀
10:16:09  rob    admin   "GET   /user/hank"
11:14:21  lucy   sales   "GET   /order/4c4712"   ◁
11:15:45  lucy   sales   "PUT   /invoice/4c4712" ◁
11:16:06  lucy   sales   "GET   /order/1d261e"   ◀
11:16:08  rob    admin   "POST  /user/hank"
11:17:35  lucy   sales   "PUT   /invoice/1d261e" ◀
11:18:22  alice  client  "GET   /invoice/4c4712" ◁
11:18:48  frank  client  "GET   /invoice/1d261e" ◀
```

**Figure 1: HTTP Log**

line represents a single HTTP request along with its `method` , `path`, `time` of operation, the requesting `user` and the `role` of the user. Following [17], if one considers attribute `method` as the operation of interest and ignores all other attribute values, then the abstracted system log is the sequence:

```
<PUT, GET, PUT, GET, GET, PUT, GET, ...>
```

This is not a very useful indicator as a model of system behavior. For example, using it to construct a database of n-grams [17] of length 3 would result in n-grams `<PUT, GET, PUT>`, `<GET, PUT, GET>`, `<PUT, GET, GET>`, and so forth. This does not capture particularly interesting system behavior and it is questionable whether it could help distinguish anomalous behavior. One reason is that the `method` attribute on its own is not sufficiently descriptive to characterize system operation. The method `PUT` may, for different events, mean making an order, creating an invoice, and so forth. Including another attribute—first part of the path (`path1`)—results in a somewhat more descriptive sequence of operations:

```
<put.order, get.userlist, put.oder, get.user, get.order, ...>
```

with resulting n-grams (n=3):

```
<put.order, get.userlist, put.oder>
<get.userlist, put.oder, get.user>
<put.oder, get.user, get.order>
```

However, this remains not a particularly effective characterization of system operation for our purposes. In this case it is a coincidence, rather than a characteristic of the system, that operations `put.order`, `get.userlist`, `put.order` appeared in this particular order. If users performed their actions at a different time, this arrangement could have been different.

A closer analysis of the log reveals that it includes a number of repeating transaction-like sequences. For example, a sequence of events indicated with ◁ is a transaction in which customer `alice` makes an order `4c4712`. A merchant `lucy` processes the order and issues an invoice which is then downloaded by `alice`. The sequence indicated with ◀ shows a similar transaction for customer `frank`. Both transactions, may be represented as a sequence of four operations

```
<put.order, get.order, put.invoice, get.invoice>
```

invoked twice, instantiated by different users and order identifiers. The above sequence is a pattern of system's behavior and is what we refer to as a *behavioral norm* [31]. It captures the essence of activity while hiding 'local' parameters (such as `time` or `user` identifiers that are not useful for this purpose. Note that including additional attribute `role` results in a more precise behavioral norm:

```
<client.put.order, sales.get.order, sales.put.invoice,
    client.get.invoice>
```

Analyzing a larger portion of this application log could result in finding other norms, representing other transactions in the system, such as user registration, returning items, managing users, posting reviews, and so forth. For example,

```
<client.put.order, sales.get.order, sales.put.invoice,
    client.get.invoice>
<client.get.signup, client.post.signup, client.confirm_email>
<client.put.return, sales.get.return, sales.post.invoice,
    client.get.invoice>
<admin.get.userlist, admin.get.user, admin.post.user>
<client.get.order, client.put.review>
```

In practice, the behavioral models that one builds from logs of past system operations will always be approximations for future acceptable behavior and, therefore, any check against these models must be able to tolerate small perturbations in the order of operations. In previous research [31] we use n-grams to model behavioral norms, enabling sequences to be matched according to a defined degree of similarity.

In order to discover behavioral norms from a system log it is necessary to first classify how log event attributes are relevant to the construction of the behavioral norm. In the above example the norms were constructed in terms of sequences of `role.method.path1` operations that are instantiated by `path2` values while ignoring `time.user` values. A search process has been developed [31] that can be used to find the most suitable classification of event attributes for norm construction from a system log. The norm discovery process has been evaluated using real system logs [31]. For example, in one experiment, norm search was performed on a low-level trace of a Java Virtual Machine hosting an enterprise application. The search identified repetitive patterns of low-level events (such as reading a file or accessing network) corresponding to high level client requests (such as sharing a file with another user). The search process identified which log attributes can be used to build suitable norms, along with other parameters, including n-gram length and similarity level for approximate norm matching. It also identified possible alternative solutions. For example, that a combination of thread, user and time attributes can be used as an alternative to an explicit transaction identifier attribute.

The applicability of behavioral norms to high level logs has also been considered. In [32] we demonstrate how norms may be used to capture and distinguish interactions between cloud service providers and consumers, as well as multiple collaborating service providers.

Norms, inferred from system logs, capture its *actual* behavior. Norms in a system may result from interactions with other systems, systems configuration or the ways that systems are interconnected. They may cover unanticipated ceremonies of interactions between a system and its users, while others may not relate to user actions at all.

# 4. TOWARDS A NORMATIVE SECURITY PARADIGM

Ensuring social order and computer security share common challenges. In this section we explore how current theories about social norms can provide insight into using normative behavior/behavioral norms to help achieve security.

We conjecture that a well-functioning system should exhibit a reasonable number of distinct norms, while in a 'broken' system, this would be not be apparent. Intuitively, a well-functioning system should demonstrate the execution of a finite number of repeating processes, and its behavior,

therefore, should not be 'chaotic'. A malfunctioning system may contain a very small number of generic norms, or a large number of specific norms that rarely repeat.

While a well-functioning system can adapt over time, this process would in itself be normative. Thus, change would be in response to new circumstances, and be an open, well understood and logical process. For example, in the successful development of norms, information may have to be perceived to be consistent with expectations. Norms initially arise to deal with specific circumstances, providing stability and predictability, and as such, are inherently resistant to change. They endure as long as those circumstances prevail, and change with changing circumstances [22].

The corollary to social norms being a means to guide our behavior, and thus providing a framework for both understanding and participating in society, is the condition that prevails in the absence of norms. Sociologist Durkheim coined the term 'anomie' to describe what this means for the individual. This state may stem from a failure to internalize the norms of a society, or an inability to adjust to changing norms. Whatever its cause, the outcome is a moral malaise with regard to norms to guide human conduct. This leads to a state of uncertainty or perhaps chaos, where what is acceptable is arbitrary, and is subject to sudden change. Think of the disintegration of society as portrayed in William Golding's novel, *The Lord of the Flies*. For most human beings, the comfort that is provided by understanding our environment and knowing that we are capable of successful participation, is the antithesis of Golding's unpredictable and savage chaos.

In addition to representing normative user interaction with the system, behavioral norms can be used to describe, normative, repeating patterns of behavior within a system itself. Our position is that system operation that is predictable and orderly is a source of orderliness. Leaving aside the technical issues around discovering norms, the question is how would being able to conceptualize system behavior in terms of norms be useful. In this broader system context, this new paradigm has wider application for assessing system security.

## 4.1 Deviations around a norm

Social norms provide a range of behavior that is acceptable in a particular situation. The middle position, as noted above, is perceived as correct [6]. Deviation from the range of acceptable behavior provokes a response that can range from ridicule, social disapproval and censure. The response depending on the norm that is violated, whether it is a common practice, such as joining a queue, or a criminal law prohibiting assault. Similarly, behavioral norms may be useful to detect anomalies in the system.

Behavioral norms can capture normal behavior as a number of distinct behaviors. An activity may be compared to existing norms and, if it does not match any of them, it can be considered an anomaly. For example, a representative norm in Section 3.2 captured customer/merchant interaction related to making an order and issuing the corresponding invoice; an activity in which a purchase order is processed without an invoice, or an invoice is issued with no order, may be detected as abnormal.

Where the norm relates to a core aspect of the group, for example, group life or loyalty, the range of acceptable behavior is narrow, however, where the norm relates to more peripheral aspects of the group, the range of acceptable behavior is wide [36]. Examples of what is acceptable being restricted would be military dress code, contrasted to a wider dress code for university lecturers. A wide range of tolerance around what is normal would allow for variance without unnecessarily suggesting a system attack. What we want is to be comfortable with a range of activities, around a norm, and to be able to identify situations or sets of circumstances where we ought be uncomfortable, and thus be able to take steps to prevent real problems escalating, or attacks on systems succeeding. Thus, imprecision and leeway per se become not only acceptable, rather they become a requirement. In this way, developing appropriate tolerance for a range of activity provides us with comfort. Behavioral norms, by providing an approximation of a system's operation, allow some degree of flexibility and tolerance within established behavioral patterns.

## 4.2 Ensuring compliance

Normative influence means that we conform to the positive expectations of others [10]. By doing so, we gain social approval, or avoid social disapproval. This comes into play when a group is perceived to have power and the ability to mediate rewards and punishment, contingent on behavior. If we belong to the group, then the norms of the group are relevant standards for our behavior. A person may adhere to norms publicly, while privately not accepting the norms, and performing actions that violate those norms. The perception that power is located in the source of the social influence is important in adhering to norms that are not internalized [28]. Absent social disapproval, compliance drops [10]. Coercion must be exercised, or a reward offered in order to achieve compliance [34]. If a norm is internalized—meaning that it is privately accepted—the changed behavior persists, in the absence of surveillance by those perceived to have power. Successfully changing norms so that they are internalized, is dealt with below.

The traditional way of ensuring compliance in computer security is through security controls. As was pointed out, this approach has its limitations. Perhaps mechanisms that provide for social disapproval could be developed and adopted, in order to assist traditional security controls. Computer systems, when a misbehaving system is detected, may react in order to change the behavior of the deviant. One way to react could be alienation. If a computer system misbehaves, it may be excluded from collaboration. To some degree, this concept is being used to fight spam. If a client/server is identified as sending spam, it may be put on a black list, which will cause other systems to reject the email it sends. To parallel the social world, where the response to violations of a norm can depend on the perceived degree of the violation, or whether the desired outcome is future compliance with the norm, or a cessation of contact, more subtle solutions exist. A 'gray' list will cause a temporary deferral, while 'yellow' enforce additional checks on incoming email [30].

Another approach to ensure compliance could be to 'strike back' [42] and take a direct offensive action against the non-compliant system. This may prevent the deviant system from impacting other systems or discourage non-compliance in the future. In society, using aggression to ensure compliance is not common and generally prohibited. It is, however, acceptable in some limited and controlled circumstances, such as using self-defense in the context of physical attack.

Also, it is effective only if the attack on the non-compliant party is successful and associated with their initial violation of the norm. What if it fails and the enemy strikes back to a greater degree? Perhaps rewarding good behavior, rather than punishing misbehavior, may be effective and less controversial. If a system may benefit from behaving in a desired way, there could be an incentive for it to adapt to what is preferred. Service providers or software vendors may tie additional functionality or improved performance to intended behaviors. For example, using more secure communication protocol may give access to more services.

In the context of users, as part of the system, social mechanisms can be used directly to increase awareness and incentives for desired security behaviors. Lipford and Zurko [25] propose that a *community oversight* can be built into the system to promote such behavior.

## 4.3 Group membership

Norms also function for groups, they co-ordinate the actions of members towards the fulfillment of group goals. This is one of the reasons for joining groups. Symbols are important in gangs, to differentiate themselves. An example would be clothing, or insignia [36]. Gangs have rigid rules for dealing with outsiders, however, leaders in the gangs exercise latitude in the norms.

In computer security, systems are often grouped according to their role or type. Depending on a group, they may need to comply with different security requirements. The criteria for grouping systems and security requirements for each group are often prescribed by security standards. For example, PCI DSS [29], requires that servers are performing only one primary function. It also specifies particular requirements for systems that store cardholder data, such as their location in the network, audit procedures and so forth.

In addition to formally prescribed groups, other emergent groups may be identified by analysis of norms. Systems that share behavioral norms may be considered as members of the same 'normative group'. For example, systems of the same role, such as HTTP servers hosting application described in Section 3.2, may be expected to be in the same group. It may raise a concern, if a system that is expected to be a member of a group, shows behavior different from its peers. Such 'deviant' systems may be identified by the comparison of behavioral norms, even if the meaning of the actual norms is not established.

A large system may consist of a number of subsystems that belong to distinct groups. For example, the order processing application from Section 3.2 may be just one of many systems used by the company, an Internet-facing system responsible for handling customer orders. Another, internal application, might be used to manage inventory, pricing and so forth. HTTP servers hosting these applications will be characterized by different sets of norms and could be considered members of distinct groups. Monitoring group membership may help identifying possible problems. Observing some behavior that is specific (normal) to the internal application on an Internet-facing server, may signal a potential problem, such as firewall misconfiguration. Alternatively, an organization may try to learn patterns of sinister behavior by deploying honeypots and looking for the occurrence of such norms on their systems.

In addition to enabling classification of systems into common groups with normative behavior, norms can also be used to characterize the kinds of interactions that individuals would like to experience with systems. An individual may wish to 'join' a particular group and adopt their behavior if the group is perceived as successful. For example, Alice has a set of norms for governing secure orderly interaction with some web service. Bob is afraid to use the service since he does not understand it, and/or how to use it safely. Bob sees himself like Alice and uses her norms to govern his own interaction with the service.

## 4.4 Observing the changing norms

Determining causation is important when observing changes in norms. Identifying causation can be complex, with a number of factors interplaying. For example, changes arise based on new technologies. An example is the printing press, which changed the dissemination of information, making printed material widespread, contributing to changes in society, for example, the Reformation. The technology per se may not have been entirely instrumental, however, it played a part in the process. Another example is the use of the telegraph, which changed the speed at which information became available. For example, news of crop failures or ships sinking, thereby influencing the markets [37].

If a social norm is operating well we feel comfortable given the sense of order, and similarly, if a system is secure, we feel comfortable and secure in its orderliness. The identification of even minor changes to the norm may give rise to a sense of unease. In an earlier section, the need to have a comfortable spectrum of acceptable behavior within society, and within computer systems, was noted. If the change identified is such that unease escalates, then we are prompted to pay attention to what is happening.

Similarly, if we're uncomfortable with the system, it draws our attention. This may indicate that a change in how it operates is being initiated. Being able to identify if this is the case is useful. If we know that the system is insecure, then the process by which this has been achieved can help us remedy the situation. For example, a routine software upgrade or adding a security control may not be intended to change a system's normative behavior. However, if it does, it might be interpreted as an unexpected side-effect of the change. This way, norms can act as a mechanism to detect common system degradation due to changes. For example, [31] shows how system misconfiguration may be detected by simple norm analysis. In the experiment, we disabled a system's access control checks to simulate unintentional configuration error. This resulted in a significant increase in the number of behavioral norms, easily identified and signifying a potential problem.

In addition, if systems controlled by different parties collaborate, the change in one system may affect others. For example, in [32] behavioral norms are used to model behavior between two collaborating cloud service providers and their user. The resulting normative behavior is sufficiently rich to enable detection of potentially malicious activity of one of the providers despite being within the scope of available security controls.

Similarly, if computer systems integrate, their behavior may change. Analyzing behavioral norms may help to measure the scope of the change and its effects on other parts of the system, not directly involved in collaboration or critical to the system. For instance, take the case of a company with a set of norms that represent its normal interactions

with its systems and services. The company decides to outsource some of these services, for example, to the cloud. As a result the set of norms of the company change to reflect normal outsourced service interaction. This set of cloud norms in the company grows as more services are outsourced. The company may be concerned that the changes due to outsourcing may give rise to a security mono-culture [18].

## 4.5 Self Improvement

Norms may change and evolve due to anomalies or small acceptable perturbations in a system, and/or user behavior. However, occasions arise where there is a deliberate intention or a requirement to change norms. One example is the campaign against smoking. In Ireland, a successful smoking ban in public places was introduced by legislation. Despite expectations of flouting, this did not occur. The social consensus in support of the ban resulted in the norm change proceeding. A less successful attempt is the enforcement of a ban on driving under the influence of alcohol, a practice that persists. Another example is that personal computers were meant to reduce paper usage, however the opposite effect resulted because it became easy to print things, often in multiple copies.

Similarly, the observation of behavior may be used to verify if changes that are meant to change system behavior are, in fact, effective. If a system has known security weakness, that is addressed by a change, such as security control, it may be expected that, after applying the change, norms also change. If they don't, the effectiveness of the mechanism may need to be reassessed.

If our goal is to cause a norm change, what are the efficient ways to achieve that? In society, minority influence can produce change in majority norms. The process, the essence of social change, is successfully achieved as follows [22]:

- disrupting the majority norm; thereby creating uncertainty and doubt
- drawing attention to itself (the minority) as an entity
- conveying the existence of an alternative, coherent point of view
- demonstrating certainty in, and unshakable commitment to, their point of view
- demonstrating that the only solution to the conflict is espousal of the minority view

Can techniques such as minority influence also be applied to change behavioral norms to achieve better security? Can an individual, small group of users, or systems, be able to influence a change that will be adopted by the rest? The network effect, very common in computer industry, can make minorities insignificant and often ignored. Disrupting the majority norm may affect interoperability and isolate the minority. In societies, as we have seen, only some initiatives to change the majority norm are successful. Studying how successful minorities operate may help achieve a better understanding of how to adopt these techniques for computer security.

## 4.6 Scope

We present behavioral norms in computer systems through an analogy with social norms and mechanisms. To date, we considered the analogy broadly and considered any kind of social/behavioral norms that might be useful. In practice, however, relating computer system behavior to society may have its limitations. Some social norms have straight forward security interpretations. For example, a norm of *hygiene* developed as preventative measures to reduce the incidence and spreading of disease. Hygienic practices are promoted by social pressure and enforced by laws and regulations. Almost identical norms, practices and policies exist in computer systems to prevent spreading of malware. An example of this close relation is adoption of hygienic/medical terminology by computer systems, such as 'quarantine', 'infection', 'health check' and so forth. *Violence* is another social phenomenon with direct computer security interpretation. An intentional use of power in order to cause harm is considered a negative behavior and is discouraged/prevented. The analogy between society and computer systems holds where social norms with direct security interpretations are considered. For other analogies, such as dress code, it does not.

The proposed paradigm is intended to apply to systems in general, that is, security not just in human-computer interaction but also from system-system interaction and, indeed, human-human interaction. In related research we have focused on some of the technical challenges of identifying norms from system logs [31, 32] that do not explicitly consider human-computer interaction. For example, [32] considers the discovery of normative behavior in the interactions between web-services; the social norm analogy gives us a new way to interpret security in this case. Discovered normative behavior is generated as a set of n-gram profiles that, in principle, can be subsequently used not just in conventional intrusion/anomaly detection but also to monitor how systems change/evolve. The focus of this paper is not about Intrusion Detection Systems per se, rather it explores the new paradigm of normative security, which, in part, may draw on techniques from Intrusion Detection.

In classic security paradigms, the attacker's goal is to find and exploit a loophole in policy or mechanism. Under normative security, an attacker may try to learn the model of normal behavior and find a means of operating within its boundaries. Similar techniques, called *mimicry attacks* [40] have been developed for computer immune systems [17] intrusion detection. Resistance to mimicry attacks depends on behavioral norms precision and coverage but also, as presented in [40], on limitations of behavioral model, and is a subject for future research.

# 5. CONCLUSION

Drawing solely on positive law is insufficient to achieve a just and orderly society, as has been illustrated. Similarly, we argue that the use of conventional security paradigms is insufficient to achieve secure orderly systems. The perception of security in society is derived not only from legislative provisions, but also from pervasive, yet unprescribed social norms. Thus, we argue, systems have analogous normative behaviors—behavioral norms—that are learnt from system logs and are not prescribed as outcomes from conventional security paradigms.

This paper presents the starting point for our new security paradigm, which is to understand security in systems in the same way that we understand security in society. In particular, that behavioral norms should be used as an indicator

of orderliness in the system. Our previous results [31, 32] demonstrate that these behavioral norms do exist in systems and can be detected. Further research is needed to evaluate how they might be interpreted and managed in practice. In this paper we use the current explanations of social norms in society to provide an interpretation for this paradigm and to give insight on how to better manage security in systems.

This paper puts forth our position on the proposed security paradigm. While we do have some initial results on potential implementation, there is much work to be done. This will include implementation of intrusion detection techniques that build on our results in [31, 32]. We are also interested in exploring the role of social and behavioral norms in human-computer interaction, however, we would agree with [33] that using qualitative research methods to discover norms in social behavior is an onerous undertaking, therefore requires resource allocation for something that may appear intangible and difficult to justify in the short-term.

## Acknowledgements

This paper benefited from lively discussion at the workshop. The authors would like to thank Mary Ellen Zurko for feedback on an earlier draft. This research has been supported in part by Science Foundation Ireland grants SFI 08/SRC/11403 and SFI 10/CE/11853.

## 6. REFERENCES

[1] *Enterprise Risk Management-Integrated Framework, Committee of Sponsoring Organizations of the Treadway Commission (COSO)*. Jersey City, NJ, 2004.

[2] *Control Objectives for Information and Related Technology (COBIT). Version 4.0.*, 2005.

[3] DSA-1571-1 openssl – predictable random number generator. Debian Security Advisory, May 2008. `http://www.debian.org/security/2008/dsa-1571`.

[4] Adams, A., and Sasse, M. A. Users are not the enemy. *Commun. ACM 42*, 12 (Dec. 1999), 40–46.

[5] Anderson, J. P. Computer security technology planning study. volume 2. Tech. rep., DTIC Document, 1972.

[6] Asch, S. E. Effects of group pressure upon the modification and distortion of judgments. *Groups, leadership, and men* (1951).

[7] Blakley, B., McDermott, E., and Geer, D. Information security is information risk management. In *Proceedings of the 2001 Workshop on New Security Paradigms* (New York, NY, USA, 2001), NSPW '01, ACM, pp. 97–104.

[8] Boyd, J. R. The essence of winning and losing. *Unpublished lecture notes* (1996). `http://www.danford.net/boyd/essence.htm`.

[9] Davies, M. J. *Asking the Law Question*. Law Book Company, 2008.

[10] Deutsch, M., and Gerard, H. B. A study of normative and informational social influences upon individual judgment. *The journal of abnormal and social psychology 51*, 3 (1955), 629.

[11] Ellickson, R. C. *Order without Law: How Neighbors Settle Disputes*. Harvard University Press, 1991.

[12] Fitzgerald, W. M., and Foley, S. N. Avoiding inconsistencies in the security content automation protocol. In *Proc. 6th Symposium on Security Analytics and Automation* (2013), IEEE.

[13] Foley, S. A non-functional approach to system integrity. *IEEE Journal on Selected Areas in Communications 21*, 1 (Jan 2003).

[14] Foley, S. Security risk management using internal controls. In *Proceedings of the first ACM workshop on Information security governance* (2009), ACM Press.

[15] Foley, S. N., and Rooney, V. M. Qualitative analysis for trust management. In *Security Protocols Workshop* (2009), pp. 298–307.

[16] Forrest, S., Hofmeyr, S. A., and Somayaji, A. Computer immunology. *Commun. ACM 40*, 10 (Oct. 1997), 88–96.

[17] Forrest, S., Hofmeyr, S. A., Somayaji, A., and Longstaff, T. A. A sense of self for unix processes. In *IEEE Symposium on Security and Privacy* (1996), pp. 120–128.

[18] Geer, D. *CyberInsecurity: The Cost of Monopoly*. Computer and Communications Industry Association (CCIA), 2003.

[19] Gollmann, D. Challenges in protocol design and analysis. In *Computer Security in the 21st Century*, D. Lee, S. Shieh, and J. Tygar, Eds. Springer US, 2005, pp. 7–24.

[20] Hamed, H., and Al-Shaer, E. Taxonomy of conflicts in network security policies. *Comm. Mag. 44*, 3 (Mar. 2006), 134–141.

[21] Hedin, D., and Sabelfeld, A. A perspective on information-flow control. In *Software Safety and Security - Tools for Analysis and Verification*. IOS Press, 2012, pp. 319–347.

[22] Hogg, M. A., and Vaughan, G. M. *Social Psychology*, 4 ed. Pearson Prentice Hall, 2005.

[23] Hutchinson, A. *Dwelling on the Threshold*. The Carswell Company, 1988.

[24] Johnson, M. L., Bellovin, S. M., Reeder, R. W., and Schechter, S. E. Laissez-faire file sharing: Access control designed for individuals at the endpoints. In *Proceedings of the 2009 Workshop on New Security Paradigms Workshop* (New York, NY, USA, 2009), NSPW '09, ACM, pp. 1–10.

[25] Lipford, H. R., and Zurko, M. E. Someone to watch over me. In *Proceedings of the 2012 Workshop on New Security Paradigms* (New York, NY, USA, 2012), NSPW '12, ACM, pp. 67–76.

[26] Mantel, H. Information flow and noninterference. In *Encyclopedia of Cryptography and Security (2nd Ed.)*. Springer, 2011, pp. 605–607.

[27] McGregor, A. Sweet justice. *Sydney Morning Herald* (March 20, 1993), 39.

[28] Moscovici, S., and Lage, E. Studies in social influence iii: Majority versus minority influence in a group. *European Journal of Social Psychology 6*, 2 (1976), 149–174.

[29] Payment Card Industry Security Standards Council. *Payment Card Industry Data Security Standard (PCI DSS)*, 2010.

[30] Perkel, M. A new paradigm for dns based lists. Available at `http://marc.info/?l=spamassassin-users&m=129364285723721&w=2`.

[31] Pieczul, O., and Foley, S. N. Discovering emergent norms in security logs. In *Communications and Network Security (CNS - SafeConfig), 2013 IEEE Conference on* (2013), pp. 438–445.

[32] Pieczul, O., and Foley, S. N. Collaborating as normal: detecting systemic anomalies in your partner. In *Security Protocols Workshop* (2014), Lecture Notes in Computer Science, Springer Verlag.

[33] Pieters, W., and Coles-Kemp, L. Reducing normative conflicts in information security. In *Proceedings of the 2011 Workshop on New Security Paradigms Workshop* (New York, NY, USA, 2011), NSPW '11, ACM, pp. 11–24.

[34] Raven, B. H. Social influence and power. In *Current studies in social psychology*, I. E. Steiner and M. E. Fishbein, Eds. Holt, Rinehart and Winston, New York, 1965.

[35] Schneier, B. The process of security. *Crypto-Gram Newsletter* (May 2000).

[36] Sherif, M., and Sherif, C. W. *Reference Groups*. Harper and Row, 1964.

[37] Standage, T. *The Victorian Internet: the remarkable story of the telegraph and the nineteenth century's on-line pioneers*. Walker & Company, 1998.

[38] Stoneburner, G., Goguen, A. Y., and Feringa, A. SP 800-30. risk management guide for information technology systems. Tech. rep., National Institute of Standards & Technology, Gaithersburg, MD, United States, 2002.

[39] Vosniadou, S. Analogical reasoning as a mechanism in knowledge acquisition: A developmental perspective. In *Similarity and Analogical Reasoning*, S. Vosniadou and A. Ortony, Eds. Cambridge University Press, New York, NY, USA, 1989, pp. 413–437.

[40] Wagner, D., and Soto, P. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (New York, NY, USA, 2002), CCS '02, ACM, pp. 255–264.

[41] Waltermire, D., Quinn, S., Scarfone, K., and Halbardier, A. The Technical Specification for the Security Content Automation Protocol: SCAP Version 1.2. *Recommendations of the National Institute of Standards and Technology, NIST-800-126* (September 2011).

[42] Welch, D. J., Buchheit, N., and Ruocco, A. Discussion: Strike back: Offensive actions in information warfare. In *Proceedings of the 1999 Workshop on New Security Paradigms* (New York, NY, USA, 2000), NSPW '99, ACM, pp. 47–52.

[43] Whitten, A., and Tygar, J. D. Why johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8* (Berkeley, CA, USA, 1999), SSYM'99, USENIX Association, pp. 14–14.

[44] Zurko, M. E. User-centered security: Stepping up to the grand challenge. In *Proceedings of the 21st Annual Computer Security Applications Conference* (Washington, DC, USA, 2005), ACSAC '05, IEEE Computer Society, pp. 187–202.

[45] Zurko, M. E., and Pieczul, O. Increasing chosen password strength, Sept 2014. US Patent App. 13/842,097.