The Landscape of Cyber-security

David D. Clark

March 8, 2015

Version 3.1

1 Introduction

This document is an attempt to describe the landscape of cyber-security. It focuses on a narrow framing–why the technology associated with cyberspace is insecure. The central conclusion is that improving the security posture of the technology is not itself a simple technical problem, but requires an understanding of economic issues, incentives and externalities, coordination and leadership, and issues of trust and (un)trustworthy actors. This framing should in turn be considered in a larger framing that considers how the limits to cyber-security influence larger issues of national security, innovation, competitive advantage, return on investment, limits on the utility of cyber-technology, the global character of discourse and interaction and the like. Determining which of these larger issues are materially linked to the degree of security in cyber-space will in turn provide a motivation for what technical aspects of cyber-security warrant a focus of attention.

2 Defining cyber-security

Computer science tends to define security in terms of the correct operation of a system: a secure system is one that does what it is supposed to do, and does not do unacceptable or unsafe things, even when it is under attack. This approach, of course, requires the functions of a system to be well-specified. There is an old saying among security experts: "A system without a specification cannot fail; it can only present surprises."[1]

It is interesting to contrast this approach to defining security with the one that we might find in political science. A political scientist of the realist school might define security by saying that a nation is secure if it can sustain peace at an acceptable cost, or alternatively if can prevail in war. Security is not automatically equated to peace; unconditional surrender will create a state of peace, but not one of security, since the price of unconditional surrender is presumably very high. In this framing of security there is no attempt to define what "correct operation of the system" would mean; that would be nonsense with respect to a nation taken as a whole. It is a pragmatic decision of the leadership whether the costs of peace are lower than the costs of war. The military exists both to deter war and prevail at war.

Another interesting contrast between these two conceptions is that one is absolute, and one is relative. Computer science tends to contemplate security as an absolute measure: the extent to which the system does or does not do what it is supposed to. In contrast, security as a political scientist would contemplate it is relative. If my nation acquires 50 spiffy airplanes, and my enemy acquires 50 spiffy airplanes, then we are probably both more secure absolutely but not more secure relatively.

Words such as cyber war and cyber weapon tend to suggest that there is an analog between conflict that occur in cyberspace and conflict that occurs in a kinetic war. This analogy is badly flawed. If one nation wants to use their airpower to destroy the critical infrastructure of another, that nation must first encounter

[1] I cannot determine who first said this. I have questioned a number of elders in the field, all of whom agree that they said it, but believe they got it from someone else.

and defeat the defenses of the enemy. Some of these defenses serve only in that context (surface to air missiles, for example), but others play a dual offense/defense role. Fighter planes exist to fight other fighter planes. If one nation uses a cyber weapon to attack critical infrastructure of another, it does not first have to fight and vanquish offensive cyber weapons of the enemy. It has to penetrate the defenses of the enemy, but cyber-technology used for defense and cyber-technology used for offense seem to have little to do with each other.[2] [3]

In this respect, the concept of cyberwar may be ill-conceived. It is the declared policy of the United States that an attack launched at us using a cyber-weapon will be judged by its consequences, and our response will be scoped based on those consequences, not whether it was a cyber-weapon. A cyber-response and a cruise missile are equally appropriate responses to a cyber-attack.

Viewed through these analogies, the computer science conception of cyber-security is curiously flawed. Most of what we see as cyber-attacks today are directed not at state assets (or in particular military assets) but the private sector. It is the private sector that must put in place the cyber-defenses. The private sector is (at least in the U.S.) prevented by law from offensive response–it cannot choose war as an option for security. At the same time, most experts agree that cyber-conflict is "offense dominant". So, in political science terms, we are trying to achieve security in a offense-dominant regime where we do not let the targets of the attack use offense, and we do not (using instruments of the state) provide that response on their behalf. In this framing, we should not be surprised if we perceive our current state of cyber-security as poor.

3 Specifying correct operation

While the proceeding discussion may hint that we should not take the computer science framing of security as the only useful one, we need to understand how to operationalize the computer science approach–how to specify the "correct operation" of a system, so as to describe the extent to which this behavior can be preserved in times of attack. The traditional computer science structural approach of layering is helpful here. Cyberspace is built up out of layers, each of which serves as a platform to provide services to the layer above. So the specification of a system normally proceeds up through the layers, asking in turn of each layer whether its service is well-specified, and whether is it "secure", which would mean that the layer will continue to fulfill its service commitment even while under attack. One can also ask whether the services of each layer are designed to make the security challenge of the next layer above it easier.

3.1 The Internet as a example

It is the connected character of cyberspace that defines it. Without connectivity, we might have lots of computers, but we would not have cyberspace as we experience it today. So the security of the Internet is central to the security of cyberspace, and the Internet makes a good case study of how to reason about cyber-security.

3.1.1 The physical layer

The physical layer of the Internet is made up of links, routers, servers, and the like. Routers and servers are computers, and thus potentially susceptible to remote attack using cyber-tools. Links themselves seem more immune to this sort of attack, and are mostly susceptible to physical attack based on close access–cutters and explosives. Links can be hardened against attack (both against destruction and tapping), and routers can be placed in physically secure facilities.

The functional specification of this layer, as we normally conceive it, is rather weak: these components are expected to do what they are designed to do except when they don't. We know that links can fail, routers can crash, and so on, and it would be foolish to pretend that we expect these components to be completely

[2] This is a thesis to be explored. To what extent are cyber-security technologies dual offense/defense?

[3] Perhaps a good analog to the framing of cyber-security today might be Switzerland, which projects no military offensive capability but a fierce defensive capability.

dependable. But given this weak specification, how would we think about the security specification? I return to this question below, since the security analysis of this layer resembles the analysis of the layer above.

3.1.2 The packet transport layer of the Internet

The Internet is a global collection of links and routers, which serve to forward units of data (called *packets*) from a entry point to an exit point. The exit point is defined by an address that is specified in the header of the packet. While there are many details about what the Internet does, at its essence this is the functional specification. And the functional specification is again very weak. The service model has been called "best effort", by which is meant that the network is expected to do its best, but failure is accepted. The network may fail to forward a packet, deliver packets out of order, deliver them multiple times, deliver them after inexplicable delays, and so on. There is a well-understood conception of what "good service" would mean–acceptable levels of loss, delay and so on, but there is no hard and fast specification. The reason for that, in the minds of the early designers, was clear–a poor service is better than none. Designers should be held to a high standard of doing well at "best effort", but there are circumstances where best is not very good. If that situation were deemed "out of spec", then those times would be considered times of failure. However, there may be applications that can still make use of whatever function there is. So this weak specification is provided to the application designers, who then have to decide how much effort to put into adapting and compensating for circumstances where "best effort" is not very good. Some applications such as real time speech that depend on good packet forwarding service may themselves not function, or even attempt to function, when the packet forwarding is functioning poorly. Others, such as delivery of email, can struggle forward even if most of the packets are being lost. The higher layers just keep resending until eventually the data gets through.

In a system like this, each layer has to take into account the failure modes of the layer below in its own design. The Internet layer is designed to take account of link and router failures–it includes a dynamic routing scheme that finds new paths if a path fails. The end to end packet transport function includes a software module at the ends of the connection that implements a protocol called Transmission Control Protocol (TCP), which copes with packet loss in the Internet. TCP numbers the packets, keeps track of which are received and which are lost, resends the lost ones, gets them in correct order, and then passes the data up to the next layer. So the overall resilience and function of the system is based not on precise specification but on a pragmatic balance of effort and investment at each layer. The better each layer, the less the layer above has to do, and (probably) the better the resulting behavior is. Different parts of the Internet can be engineered to different levels of performance and reliability (driven in many cases by pragmatic considerations of cost), and each layer above is expected to cope with this variation. Investment at the lower layers benefits all of the next layer functions, but over-investment at the lower layer may add unnecessary cost to the service. None of this is part of the Internet's "specification"; the interplay between performance and reliability at the different layers is a point of constant adaptation as the Internet evolves.

In this context, how would we characterize the "security" of the packet forwarding service of the Internet? A formally correct but useless response would be that since the network is "allowed" to fail, it need not concern itself with security. Pragmatically, of course, this is nonsense. There are well-understood expectations of the Internet today, and an attack that materially degrades that service is a successful attack. But it is a matter of degree. Degraded service may still be useful.[4] But with a loose functional specification like this, the determination of how to make the system resistant to attack is potentially ad hoc. One must look to the design mechanisms, not the specification, to see where attacks might come. Thus, one would look to the routing protocols, and ask if they are robust to attack (they are not, as I will discuss below). But the core function of the Internet is actually very simple. If there are links connecting routers, and the routers are working, and the routing protocols are computing routes, the Internet is mostly working.

The next question is what services the packet forwarding layer provides to make the "security job" of the next layer and the other components at the same layer easier. The answer is "not many". This might

[4] Security experts understand that the most dangerous attacks are those that might cause massive, correlated failure of components, for example attacks on routers that exploit a common failure mode and take out so many routers that the dynamic routing algorithms of the network are overwhelmed and the network essentially ceases to function.

be seen as an error, and a historical perspective may be helpful in understanding this state of affairs.

4 The early conception of "Internet security"

Some of the early Internet architects, including me, have been criticized for not thinking about security from the start. This criticism is to some extent valid, but in fact we did consider security; we just did not know at that time how to think about it. We made some simplifying assumptions that turned out to be false.

The starting point for many discussions about computer security divide the problem into three sub-objectives: confidentiality, integrity and availability (CIA). These objectives make the most sense at the level of information security–information should not be disclosed except to parties authorized to see it, information should not be corrupted, and it should be available. This framing separates the world into two sets of people–those who are authorized and those who are not. If an actor is authorized, then they can see the information, and if they modify it, this is not corruption, just modification, since they are authorized. If an actor is not authorized, then the goal of the system is to deny them access.

This framing is deceptive, but it shaped our early thinking. We knew that some routers might be suspect, so there was no way we could insure that a router did not make a copy of a packet–the packet forwarding layer could not itself provide confidentiality. And a malicious router might modify a packet–the packet forwarding layer could not itself provide integrity for data in transit. We took a very simple view, which is associated with a mode of thinking called "end-to-end": only the end points could undertake to mitigate these vulnerabilities and achieve these objectives because only they could know what the objective was, and only they were (presumably) trusted and authorized to exchange this data. End-to-end encryption is the obvious approach: if the data is encrypted making a copy is useless and any modification can be detected. This approach leaves only availability for the network to solve. Of course, "all" the network does is deliver packets, so it would seem that availability is the core requirement. In this context, it is interesting that we have no "theory of availability", but I defer that consideration until later.

Why was this CIA framing deceptive? It implies a simple world model–mutually trusting parties communicate and parties that do not trust each other do not. It concerned itself only with information security among mutually trusting actors. What we missed was that most of the communication on the Internet would be between parties that were prepared to communicate but did not know whether to trust each other. We agree to receive email knowing that it might be spam or have attachments that contain malware. We go to web sites even though we know (or should know) that web sites can download malware onto our computers. This is the space we need to make secure, not just the space of CIA communication among known, trusting parties.

In this context, the end-to-end principle is not wrong, just incomplete and in need of re-interpretation. An analogy may help. If trusting parties want to send a private letter, they want assurances that the letter is not opened in transit. But if recipients suddenly realize that they may get a letter full of anthrax, then their "security objective" reverses–they want that letter opened and inspected by a trained, trustworthy (and well-protected) intermediary. End-to-end encryption between and attacker and his target is the last thing the target wants–it means that the target can get no help from trusted third parties in protection. An encrypted exchange with an untrustworthy party is like meeting them in a dark alley–there are no witnesses and no protections.

We cannot expect "the network"–the packet forwarding layer of the Internet–to know which of my potential communicants I trust, which flows require intermediation, and so on. That approach would not scale, and further would require that the end-node know that the trust thus placed in the Internet was itself not misplaced. So in fact the simple semantics of the Internet–best-effort delivery of packets–is (for the moment) about all the network can do. But the overall security problem is not solved by telling the higher layer to use end-to-end encryption. The problem of operation in an untrustworthy world has to be handled explicitly by the next layers in the system, the application layer, and it was this design problem which we neither clearly articulated nor explored how to accomplish. Application designers should not have to solve these problems from scratch each time a new application is designed; what is needed is advice and guidance, perhaps applicable to a class of applications, that suggests how these problems might be approached. What is needed is a collection of *application design patterns* that can be offered to designers. Trying to think about design patterns in an organized way should yield another benefit; by looking across applications to see common needs, new ideas may emerge for common services that the lower layers can offer to help improve the security of applications. It is highly unlikely that there will be some new service at the packet forwarding layer that can suddenly make applications secure, but is is possible that there are supporting services that can make the task easier. The way to find these services is to look at application requirements, generalize from them, and see what concepts emerge.

5 The landscape of security

At this point, we can begin to envision the landscape of security. There are actually two different illustrations that capture what I have described. The first is a layered picture: lower layers providing services to higher layers. Lower layers are more general: the packet transport layer of the Internet is very general–it just moves packets of data. On top of this a general application might be fashioned, such as the Web. In turn, on top of this a more specific service might be crafted, such as Facebook, and on top of this there might be specific "Facebook apps" that run in the context of Facebook. The Internet is a layer-cake of platforms, each supporting services on top of them.

At each layer, the security analysis must include an analysis of the extent to which untrusted parties are either intentionally or unavoidably in the system, whether their actions can cause harm, and if so how to discipline those behaviors. In general, this responsibility cannot be "pushed down" into a lower layer, but it may be possible for the lower layer to provide supporting services to make that task easier.

The other illustration is a regional one, much like a map of the globe. We know that there are regions of the world we don't much trust. The same is true of the Internet; there are regions of the packet forwarding layer of the Internet (they are called *Autonomous Systems*, or ASes) that are not trustworthy, and regions can and do attack each other (in particular the routing protocols). And this regional structure is true at every layer: there are untrustworthy email senders, untrustworthy web sites, and so on. In the context of certain applications, if we can detect them we can try to eject them from the community of mutually trusting actors (this is what anti-abuse institutions such as Spamhaus try to do with spammers) but in general we have to accept that they are in the system, and must be tolerated, if not welcomed.

5.1 Mapping classic "security problems" into this landscape

It is useful to take these illustrations of the security landscape and use them to try to position some of the well-known categories of "security problem".

5.1.1 Attacks on the network itself

These include attacks on the routing protocols, attacks on critical supporting services such as the Domain Name Service (the DNS), and the like. Since the core function of the Internet is actually rather simple, there are only a few of these services; the interesting question is why they remain insecure. I return to this below. To the extent that this layer cannot detect and remedy these problems internally, the consequences of attacks at this layer will become visible to the layers above, which will have to take corrective action.

5.1.2 Attacks on the attached hosts

Attacks on attached hosts can occur as a result of communication with a malicious party (who uses the capabilities of one or another layer to deliver an attack) or as a result of an unsolicited incoming packet that somehow exploits a vulnerability to launch a successful attack. Over the years, these sorts of attacks have been "moving up the layers". In the past, there were some well-known vulnerabilities in the software that supported the packet transport layer–for example packets that would cause problems with the TCP layer. In most implementations, these have been fixed. Most operating systems today are reasonably secure against attacks at that level, so the attacks now target application code. As that gets better, of course, the attacks target the user–attempting to fool the user into executing some unsafe action.

There has been much progress in devising ways that the operating system of the end-node, in addition to being more robust itself to attack, can help protect the application running on the end-node from attack. The concept of *sandboxing* describes an approach where the code of the application is placed in a confining environment before it interacts with the network, and this environment is conceptually discarded at the end of the interaction, thus discarding in passing any malware or other modifications that may have resulted from the interaction.

5.1.3 Attacks on the communication

This is the classic problem I described before, sometimes classified as information security, where parties attempting to accomplish mutual communication are thwarted by an attack, perhaps launched by the network or by some party that has gained control of some critical control point. A few years ago, there was a furor in the U.S. because Comcast blocked a peer-to-peer music sharing application (BitTorrent) by injecting forged packets into the data stream. This was not seen as a "security" event but as a violation of the norms of service, but in the language of security, this was without a doubt an attack on a communication by the network. End-to-end encryption would have detected this particular attack, but since this was intended to be an attack on availability of service (see below) there could have been many other approaches.

5.2 A taxonomy of actors

Another way to visualize the landscape of security is to catalog the actors, both good and malicious, and see where they fit, both into the layer picture and into the regional picture.

5.3 Individual users

The individual (or user, or consumer or citizen) has a set of concerns that are pretty well understood. One is the actual loss of data: "Where did my wedding pictures go??". I believe this happens far more often due to a disk crash on a machine that is not backed up than due to a malicious attack, but the concept of "ransomware"–malware that encrypts a disk and demands payment to provide the decryption key–is an example of loss of data due to actual attack. Consumers are also concerned with the unauthorized release of (at least) some of their data–financial information, identity information and the like, if not wedding pictures. This is sometimes stolen from individual computers, but more likely stolen in bulk from enterprises that have it stored in their servers. And again, malware on a home computer can capture and steal passwords and the like, facilitating access to personal information. Many of these harms arise as a result of malware on the individual's computer, so the security of that computer would be a good, foundational element of good overall security.

Again, the question is the extent to which the operating system of the computer (the software that makes the computer into a general platform for the applications) can provide this role, and the extent to which the applications must play a role.

Privacy is a concern of the individual, and thus fits under this heading. Loss of privacy can be due to a failure of security, but also due to behavior by legitimate actors that do not act in the best interest of their users. So privacy may be an aspect of security or an independent consideration. And of course, privacy may be seen as a barrier to national security, discussed below. The relation between privacy and security, as headline terms, is complex.

5.4 The enterprise

The enterprise is concerned with loss of data, theft of data (espionage, theft of personal data about their customers, and so on) and availability. They thus face the full suite of CIA with respect to their information. This is a complex space today–as basic defenses have gotten better, attacks have become more complex, with sophisticated attacks today involving a series of steps, each of which must bypass or circumvent some protection to gain another capability in the attack process. The attacks move up and down the layers, and there is no single moment where the attack has "succeeded" until the final step that represents the actual harm. In the case of many attacks, the final success may depend on actions far away from the attacked system: to exploit credit cards the attacker must not only steal them but then sell them, which may involve overseas transactions with other untrustworthy actors. Crime today has a complex production chain. But, again, the vectors of initial entry have been moving "up the layers", and today often begin by misuse of an application to mislead a user, such as a phishing email. In this context, it becomes clear that one cannot talk about "making the Internet more secure" as if it were a monolithic whole.

Distributed Denial of Service attacks

A problem of particular concern is an attack on availability by flooding a server (or the network immediately serving it) with so much traffic that valid traffic cannot get through. These attacks are called Denial of Service attacks, and since they are often launched by using a large set of attacking machines to launch the attack, they are called Distributed Denial of Service (DDoS) attacks. From the point of view of the victim, they are frustrating and hard to mitigate, although an industry has sprung up to help provide this service. From an analytical perspective, they are interesting due to the method the attackers commonly use to obtain these sets of attack machines: they obtain them by inserting malware into otherwise innocent end-nodes on the Internet, and then instructing this malware to carry out whatever sort of malicious behavior is desired. Machines thus taken over are called *bots*, and the resulting collection a *botnet*. Why can this situation persist? Why are botnets so prevalent that a aspiring villain can rent time on them by the hour from their "owner"?

The technical answer is that across all the layers of software on a modern end-node there are and always will be vulnerabilities which can be exploited by the attacker building his botnet. Attackers have developed very sophisticated ways to exploit these vulnerabilities. One approach, adding one more step to the overall attack, is to find and penetrate poorly managed web sites, in order to add malicious content to otherwise innocent web pages. Then the so-called *botmaster* just sits back and waits for end-nodes to make contact with these web sites, which will trigger attack by this content, which in turn causes the end-node to be infected with malware.

Viewing this situation not as a purely technical problem but more generally, what this story reveals (aside from the complexity of today's attacks) is a problem that can be classified as a negative externality. A web site manager who is lax in maintaining his web server (failing to install the latest security patches and the like) may have his site infected with malware, but this may not cause him any direct harm, unless he suffers loss of reputation or performance. The owner of the end-node, often an individual at home, has very little interest in becoming a security expert. If his system becomes infected with malware that turns his machine into a bot, it may not seriously affect him if the bot is designed not to draw attention to itself by causing massive performance problems. Removing the malware would just cost the owner of the machine time and frustration. Why bother? So malware persists on the Internet, because the cost of that malware is not carried by the owner of the machine who is hosting it but by the ultimate victim of the attack launched from it.

Of course, policy could shift the landscape of security here. Instead of accepting the cost of tolerating this malware (the cost of peace) we could go to war against it. We could instruct ISPs that when they detect a machine infected with malware they disconnect it from the network. ISPs do not want to be given this duty, both because it leads to bad relations with their customers, and because it is the help desk of the ISP that the customer will call, generating costs for the ISP for which they may not be compensated. This space is a cascade of misaligned incentives and negative externalities.

5.5 The state

The state has a range of concerns. One might try sorting them into direct security concerns, economic concerns, and concerns over the protection of citizens. The term *national security*, which is certainly one of the legitimate conceptions of security, centers on threats to the state, especially those that have risen to a certain level of concern: existential threats, issues of *high politics* and the like. Attacks on a state's military capacity or critical infrastructure fall in this category. Such attacks can fall in any dimension of CIA–that categorization may not be the most useful in determining the severity of the threat. Economic issues can include specific attempts to destabilize or degrade an economy, or espionage that causes loss of competitive advantage of the private sector relative to other states. Concerns over the protection of citizens can be highly variable; one nation may protect the right of free speech, another protecting its citizens from exposure to unacceptable content. In some cases this latter concern may rise to the level of national security, if access to unacceptable content is seen as regime destabilization.

5.6 The global system

The global system, which can be viewed through many lenses–anarchy shaped by realist exercise of power, a space shaped by global institutions and so on–has its own concerns, even if they are not often or not clearly articulated. Global connectedness may be a great force for stability, and attempts to disrupt openness may be seen by some players as an attack on global stability.

What we see overall is that these concerns do not map onto the landscape of layers–with a few specific exceptions (ISPs live at the IP layer and tend to launch attacks at that layer) attackers are nimble at moving among layers to achieve their goals, which implies that one cannot assign the responsibility of good security to a single layer–there is no *security* layer in the architecture. The analysis above, which points out that there are security tradeoffs among the layers, some fundamental, some a matter of cost, is essential to understanding how to achieve improved security. And this fact in turn captures one of the key issues with respect to security. Different layers tend to be under the control of different institutions and actors. Organizations concerned with standardization, operation and governance tend to align with layers, so resolution of security problems requires cross-institution conversation, not just cross-layer technical design.

Some of these concerns might, however, map onto the landscape of regions. It is possible that one might find a concentration of malicious actors in a region that shows tolerance for these actors–Nigeria is famous for its concentration of email scammers. This raises the question of whether, in some cases, one might punish a state for hosting malicious actors, just as (and with far more intensity) we punish states for harboring terrorists. Might we, for example, degrade the email connectivity from Nigeria as punishment for its lack of regulation of malicious actors? This takes us back to the calculus of political science security–do we prefer the cost of security in the context of conflict (erosion of connectivity, degraded email, possible retaliation) to the cost of security in the context of peace (global connectedness, but citizen annoyance, fraud, and the like).

6 Availability and the role of trust

The computer science security community has given us a powerful set of tools to improve security, with encryption being among the most powerful. However, it is important to see how these tools tend to function in the larger context. As I discussed above, encryption fits into the CIA triad by giving strong assurance that data is not disclosed, and strong indications if data is modified. There are several well-understood contexts in the Internet today in which encryption is deployed. These protect the user from failures of integrity by halting the communication. They map a wide range of attacks into a common outcome–cessation of communication. But this outcome, while potentially better than a failure of confidentiality or integrity, is just a failure along the third CIA dimension–availability. Essentially what these schemes do is turn a wide range of attacks into attacks on availability. And that is not the desired outcome–we want to offer assurances about all dimensions of CIA.[5]

If the best we can do using encryption is to turn a range of attacks by untrustworthy actors into attacks on availability, what can we do to improve the situation? There are two ways to try to deal with untrustworthy actors: constrain or discipline them, or avoid them. Imposing constraints on untrustworthy or malicious actors that both compel them not to misbehave and as well compel them to perform at all are hard to devise; the *fail-stop* semantics of attack detection is the common outcome. The only way to compel correct operation is to so constrain the system so that the cost to the actor from expulsion from the system outweighs the cost from foregoing malicious behavior. This might work for an ISP who is hosting both legitimate customers and spammers (and ISPs have been expelled from the Internet for hosting spammers, essentially driving them out of business), but malicious individuals show great resilience to constraint and discipline, especially across region boundaries. Thus, if we pick conflict rather than peace as the response to malice, exclusion is the normal outcome. If there is a malicious ISP, don't route through it. If there is a email sender that seems to send only spam, block receipt from it (this sort of treatment is what anti-abuse organizations such as Spamhous try to coordinate). Essentially, if we want all of the CIA triad, we must organize the system so that even if untrustworthy actors are in the system, we do not depend on them. We tolerate them if we must, but we do not make any interactions among mutually trusting actors depend on untrustworthy elements.

This realization has been slow to come into focus for some designers of security mechanisms, because it is a shift in mind-set. But this shift is the basis of what can become a "theory of availability". The necessary basis is that some mechanisms within the system must be able to detect that something is wrong, localize the source of the malfunction, and exclude it from being used. This logic is completely obvious to designers when it comes to failures: if a router has failed, the protocols must be able to detect the failure, and there must be sufficient redundant routes that a dynamic routing protocol can "route around" the failure. But if the problem is not a failure but an attack, this logic gets much harder, and it brings us back to the layered analysis. Can a layer be expected to detect all the attacks that are arising at that layer that do not affect that layer itself, but the layers above it?

Consider a very simple example–a router that drops or adds packets to a packet flow. This sort of action does not break the forwarding layer, just the end-to-end communication. Should the packet forwarding layer keep count of packets, and exchange these counts to see what is being lost or gained? The complexity and performance cost is daunting. Or consider the more subtle attack of changing a bit in an encrypted packet. This attack disrupts the higher-level flow. Should the network re-compute the encryption function at each node to detect that (and where) the packet is corrupted? Would it even have the necessary encryption keys, or be authorized to have them?

The design of the Internet is based on the quite defensible assumption that these sorts of attacks can easily be detected at the end points, but not in the network, so the task of detecting them is delegated to the end. But assuming that the end-point detects a failure, what can it do? In today's Internet, the end-points have very little or no control over network functions like routing. If communication between end-nodes is being attacked, the end-nodes have no general way to localize the problem, and no way to "route around" it. If the network design gave that sort of control to the end-nodes, those mechanisms themselves might become attack vectors, so they would have to be designed with great care. Today, what happens is that we accept that a range of attacks are going to result in loss of availability. If a network must function at high levels of availability, we use non-technical means to make sure that only trustworthy components and actors are in the system. So to achieve the full complement of CIA, both technical means and operational and management means must be combined as part of the approach.

However, at a fundamental level, improving the availability of the Internet faces a basic conundrum. If only the end-nodes can detect failures of availability due to attacks, and the end-node cannot be trusted to reconfigure the network lest this be another attack vector, there would seem to be no way to resolve such

[5] This observation provides one explanation as to why so many users deal with dialog boxes warning about potential hazards by clicking the "proceed anyway" option–what they want is to make progress. Another reason, of course, is the often inexplicable content of those warnings.

problems. Working around this conundrum is a challenging design problem that involves creation of control structures that build on trustworthy components (which would have to be specified and implemented) to provide a foundation for these sorts of functions.

7 Making systems trustworthy–expanding the landscape

The discussion to this point has focused specifically on issues of *security*, which implies the actions and consequences of malicious actors. What the user is concerned about is whether the system is *trustworthy*, which is a broader goal that implies that the user understands what the system will do, and that the user is justified in trusting that the system will do what it is supposed to, and not behave in ways that are adverse to the interests of the user. Sometimes the user's assumption about what the system will do is implicit, since today's systems are quite complex, and so the understanding that the system will perform in ways that are consistent with the user's interests and expectations is often a somewhat high-level hope not based on a precise specification. Just as we talk about the "security of the state" as an abstract concept, whether a system is trustworthy is sometimes a rather abstract concept that is sometimes only debated after what appears to be a violation of that trust. But the maxim quoted above that a system without a specification cannot fail also applies here.

Sometimes the conditions that attempt to create a trustworthy context are defined outside the system, perhaps by law. In the medical context, HIPAA defines a rule set that in the eyes of the creators would create a trustworthy system for medical informations. Obviously, two distinct questions follow: should we as users accept that a system that is compliant with HIPAA is actually trustworthy, and is a system with which we are interacting actually compliant. [Perhaps cite CSTB study on limits to preservation of privacy in HIPAA compliant systems?]

But even if the conception of a trustworthy system is, in the large, rather abstract, the previous discussions around security allow us to frame at least part of a analysis of trustworthy behavior. At the level of data transport in the Internet, the only expectation that both users and application designers are justified in having is availability. The Internet cannot make applications trustworthy, nor can it deal with loss of confidentiality or integrity of data in transit. Availability–the ability to communicate as and when needed–is its core value.

The previous discussion suggests a few ways to reason about availability that can be expanded to deal with a broader range of impairments. Interestingly, the basic design of the Internet does not speak to issues of availability. To deal with failures, there must be enough redundancy in the network to continue operation without the failed components, and there must be dynamic adaptation algorithms to bring these assets into play. But the degree and configuration of redundancy is a matter that is defined as the the network is built by its operators–the Internet Service Providers. Some may invest more, some less. Some may plan for disaster, some just hope for the best. The design of the Internet allows for this variation.

A key challenge here is that the research community does not have good methods and models to relate different sorts of redundancy to different outcomes under different failure scenarios. And if these methods and models existed, they would have to be tuned or reconstructed if the dynamic adaptation algorithms used in the Internet change, and these algorithms are themselves different in different parts of the Internet, and subject to change as the Internet evolves. What we have today is a rather pragmatic, seat of the pants engineering approach that in practice has not allowed major disruptions to Internet service, even when the system is subjected to major jolts [Cite CSTB study on 9/11.]

The more complex questions about trustworthy behavior arise at the higher layers of the system, in the applications and services with which the user directly interacts. The behavior at this level is more complex, the opportunities for malice, misunderstanding and mis-alignment of interests are much greater. Most of the applications used today on the Internet are created by commercial actors whose primary motivation is profitability. Applications such as Facebook or Twitter must be appealing to users or they would not succeed in the market (succeed in making money) but there is a tension between meeting the needs of the user and adding features that make money, which might (for example) involve selling demographic information about those users. The balance of these sorts of issues are often the subject of law and regulation, as well as a changing landscape of norms and expectations.

The language of security–of attack and defense–paints a picture where actors are either good or bad–white hats or black hats. But the issues that can erode trust in a system may arise in a context where actors have mis-aligned interests but are not necessarily breaking the law–spammers claim that their behavior is not "black hat", but legitimate business practice. In fact, in the U.S. sending unsolicited bulk mail is an exercise of free speech, and attempts to block spam have been denounced as censorship. This sort of contention serves to remind us that part of what defines the experience of using the Internet is trying to create a trustworthy experience in a context where we must accept and tolerate actors that are not mutually trusting, and who do not have aligned interests.

8 Barriers to better security

The conclusion I want to draw from this analysis is that security problems in the current Internet are not the result of lack of technology. The barriers are problems of coordinating and incentivizing collective action, dealing with negative externalities and costs imposed on first-movers, understanding how to cope with a lack of uniform trust across the system, and the like. To overcome these barriers will require good system design, but that design is not exclusively technical. There must be complementary aspects of technology, operational requirements, governance, and the like.

Comparing the "computer science" and "political science" definitions of security sheds some light on these issues. The computer science definition of security–that a system will only do what it is specified to do, even under attack–defends against unexpected outcomes or behavior but is not framed in terms of preventing specific harms. It is an appealing definition to a system engineer, because it seems to frame security in a way that bounds the problem to the system in question. Framing security in terms of preventing harms (e.g., preventing credit card fraud) brings many more elements into scope. For example, preventing or mitigating some forms of credit card fraud may best done by modification of the credit card clearing system. This definition of security frustrates the designer of a component, because the scope of the solution is no longer within the scope of the designer to fix. Of course, if the system in question is a multi-component system with multiple actors responsible for parts, even the "computer science" definition of security may be hard to contemplate.

Here is a case study that illustrates some of these barriers, and as well suggests what sort of design logic might mitigate some of these barriers. Earlier, I pointed out that some of the key functions of the Internet–the routing protocols and the Domain Name System–manifest some well-known security vulnerabilities. Why is this? The problem is well understood and the context in which the solution must be implemented is well understood. The problem is often one of collective action, and negative externalities by first movers. The box Securing Interdomain Routing in the Internet contains an extensive discussion the challenge of making interdomain routing more secure. It is an example where lack of trust and coordination problems are barriers to design and deployment of a more secure Internet.

Securing interdomain routing in the Internet

As I have described earlier, the Internet is made up of regions called Autonomous Systems, or ASes. Each AS must tell the others which addressed are located with the AS and how the ASes are connected in order for the Internet to come into existence. The way this works in the network today is that each region announces the addresses that are in its region to its neighbors, who in turn pass this on to their neighbors, and so on, until this message reaches all of the Internet. Each such message, as it flows across the global network, accumulates the list of ASes through which a packet can be sent to reach those addresses. Of course, there may be many such paths–a particular AS may be reachable via many neighbors, and so on. So a sender must pick the path it prefers, or more precisely, each AS computing a route back to a particular set of addresses must pick among the options offered to it, and then offer that option to its neighbors in turn.

Today, there are no technical security controls on this mechanism. That is, any rogue AS can announce that it is a route (indeed, a very good route) to any other AS in the Internet. What may then happen, if other ASes believe this announcement, is that traffic is deflected into that AS, where it can be dropped, examined, and so on. This sort of event, in fact, is not uncommon in the Internet today, resulting in failures along all dimensions of CIA. How is it fixed today? Network operators monitor the system, problems of reachability are reported from the edge by end-users (who often have to resort to phone calls, since their systems cannot influence routing) and over some period, perhaps a few hours, the offending AS is identified and isolated until some suitable discipline can be devised.

Ignoring details, there might seem to be an obvious technical fix. Why are these announcements not signed, using some sort of cryptographic scheme, so that they cannot be forged? Indeed, this was the path down which the designers started when they set out to secure interdomain routing. But there are two formidable barriers to this, one having to do with trust and one have to do with migration to the new scheme.

The migration problem is easy to understand. In the global Internet, there is no way that everyone is going to switch to the new scheme at once. Unless some draconian discipline is applied (disconnection from the net), some actors may just refuse to undertake the effort of upgrading, and they will continue to originate route assertions that are unsigned. There are two options to deal with this. One is to reject them (which is the draconian outcome of disconnecting them) or accept them, in which case a malicious actor cannot be distinguished from a lazy actor, and we are essentially no better off. Until the last AS converts, we get little value from the scheme, unless we wrap it in complex high-level systems, such as globally distributed, trustworthy lists of ASes that have converted, so that a router knows which unsigned assertions to accept.

The issue of trust is just a little more complex. When an AS signs an assertion (for example, when MIT signs the assertion that it is AS 3, and that it has a particular set of addresses that it holds within that domain), it has to use some encryption key to sign that assertion. The obvious technical approach is to use what is called a public or asymmetric key system, where MIT has a private (secret) key it uses to sign the assertion, and a public key it gives to everyone so they can decrypt the assertion and confirm that MIT signed it. So far so good, but where does that public-private key pair come from? If MIT can just issue itself a set of keys and start signing assertions, it might seem that we are no better off, because a malicious actor could do the same thing–make up a public-private key pair and start signing assertions that it owns AS 3, controls those addresses, and so on. To prevent this from being effective, the technical proposal was to create a trusted third party that could confirm, based on its own due diligence, which public key is actually associated with the real MIT. But why in turn would anyone trust that third party? A scheme like this ends up in a tree of trust, which seems to require a *root of trust* at the beginning, a single node we all trust to tell us which second-level parties to trust, and so on until we get to the party that asserts that it know who the real MIT is.

An engineer might think this was a simple, elegant scheme, but it runs aground in the larger world. First, what single entity in the world would all the regions of the world agree to trust? The United Nations? This issue is serious, not just abstractly but very concretely. When this scheme was proposed, several countries (including Russia) asserted that they would not assent to a common root of trust with the U.S. The agent who has the power to validate these assertions must, almost of necessity, have the power to revoke these assertions. Can we imagine a world in which the United Nations, by some sort of vote, revokes its trust assertion about some nation and essentially ejects that region from the Internet? What about those second-level entities, that almost certainly are within some legal jurisdiction and thus presumably subject to the legal regime of that region?

This fear is not hypothetical. The second level entity nominated to play this role for the EU is called RIPE, and is located in Holland. Once they take on this role, the Dutch police can bring a police order for them to revoke the certification of an AS, including one not in Holland. The Dutch police already brought such an action, but because the scheme for signing these assertions was not in place, RIPE correctly said that it did not have that power. Once the encryption is made operational, they can no longer make that claim.

So does this scheme make the Internet more stable or less? Once people understood the social consequences of this scheme, there was substantial resistance to deployment. The problem with adding a “kill switch” to the Internet is to control who has access to it.

What is needed is a different design approach, one that allows actors (e.g., ASes) to make assertions about who they are, but validates these assertions in a way that makes them very hard to revoke. That would solve the “jurisdiction” problem. But if a false assertion ever got started, how could it ever be revoked? Once we grasp the complexity of functioning in a space where not all the actors share the same incentives, not all are equally trustworthy by different measures, and that these actors of necessity are in the system, it becomes a very difficult problem indeed to design a system that is robust at ejecting actors that are “bad” but also robust at not ejecting actors that are judged “bad” if we don’t accept that they are bad. Management of trust relationship, and the expression and manifestation of those relationships, becomes the defining feature of a successful scheme, not exactly how crypto is used.

So in this respect, the landscape of security becomes a landscape of trust–regions of mutual trust will be more connected, more functional, more effective, and regions that don’t trust each other will still try to communicate, but with more constraints, more limitations, and perhaps more failures, especially with respect to availability. And this pattern will be found within any application that tries to tailor its behavior to the degree of trust among the communicating parties, whether the function is exchange of routing information or email.

What happens today is that “the Internet” does not try to solve these problems using technology. We fix some of these problems using management–oversight of the system by trained operators and managers. We just tolerate some of the consequences, such as receipt of untrustworthy mail.

An alternative to the scheme described above to secure the AS routing system will illustrate how a different scheme fits into a socio-technical architecture. The scheme described above, with a hierarchy of trusted certifiers and a single root of trust, is technically robust, in that it will always give the right answer *if the trust relations are valid and accepted by all the parties.* This approach may be technically robust but is not socially robust. Here is an alternative approach that is less technically robust (one cannot prove that if it will give the correct answer under certain assumptions) but is more socially robust. Above, I rejected the idea that MIT just make up a public-private key pair and start signing its assertion. What would happen if that scheme were adopted? At first, various regions of the Internet might get conflicting assertions, if it happened that there was a malicious actor in the system at the time when the assertions started to be signed. That situation, while not desirable, is what we have today. But over time–days or weeks–it would become clear what key went with the “real” MIT. Each AS in the network could learn this for itself, or groups of mutually trusting ASes could cooperate to learn it. The public key could be exchanged by side-channels–written on business cards or on a napkin at a bar. Once the other ASes in the Internet have decided which key to trust, they have independent possession of that fact, and there is no authority that can compel a third party to invalidate it. Any AS can decide on its own to stop forwarding traffic to MIT, just as they can today.

What this scheme exploits is not a technical scheme for propagating trust, but a social protocol called “getting to know you”, which humans have been running, probably for millions of years. We can be fooled, but in fact we are pretty good at it. And it is simple. It requires no trusted third parties, little administration (except that each AS should try very hard not to lose their own private key) and great adaptability to changes in the landscape of trust.

9 Acknowledgement

This paper has greatly benefitted from discussion with Josephine Wolff, Shirley Hung and Nazli Choucri at MIT.