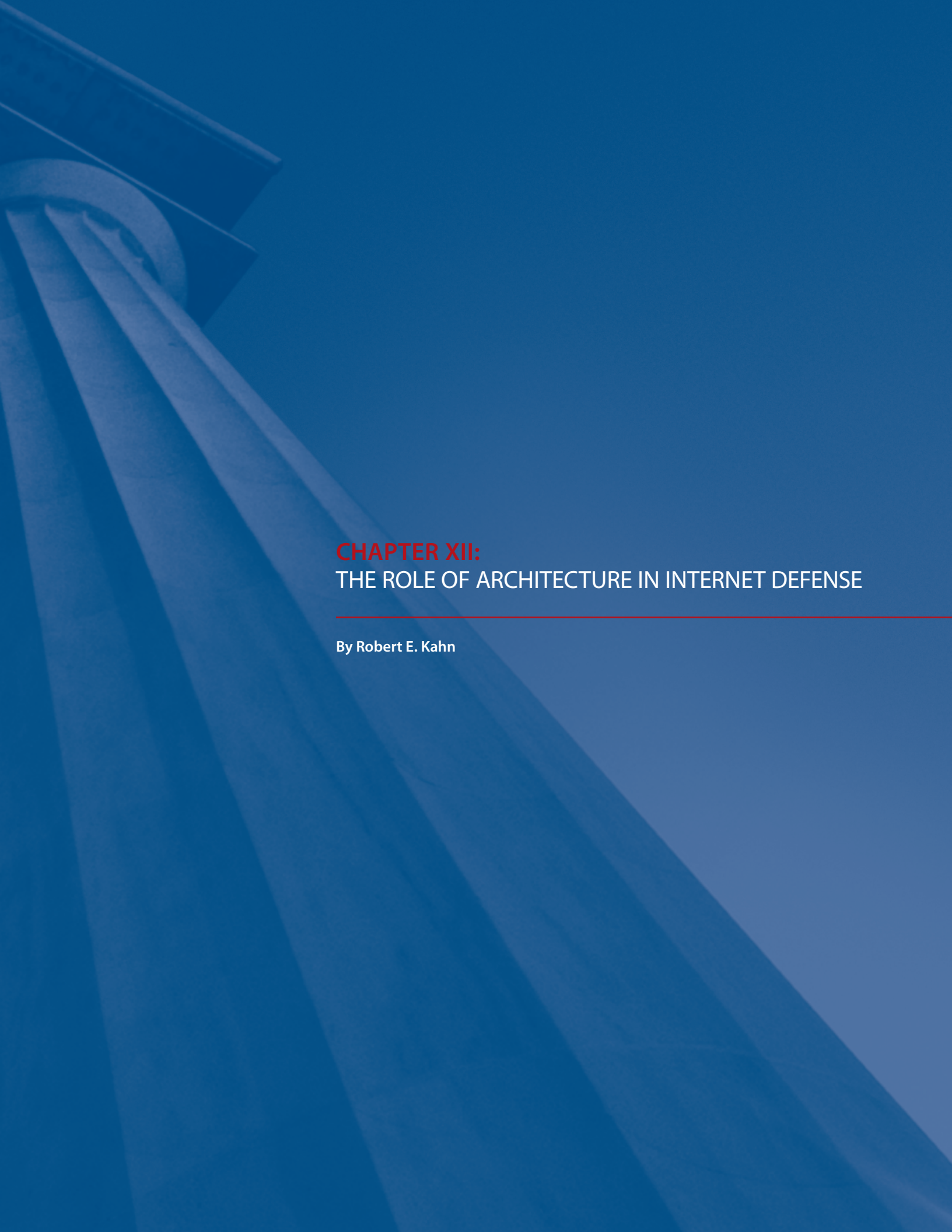

CHAPTER XII:
THE ROLE OF ARCHITECTURE IN INTERNET DEFENSE

By Robert E. Kahn

America's Cyber Future

Security and Prosperity in the Information Age

THE ROLE OF ARCHITECTURE IN INTERNET DEFENSE

By Robert E. Kahn

Since it was first introduced in the early 1970s, the Internet has met the growing needs of an ever-widening community of users with great benefits to individuals, organizations, governments and their associated disciplines. Yet, along with that growth and evolution has come an increasing downside, namely traffic that intrudes and may disrupt productive uses of the Internet. Worse yet, concerns exist that such unwanted and unwarranted intrusions may cause more extensive damage in the future. Managers of information systems and resources attempt to find ways to ensure that access controls are not breached, or that intrusions or disruptions have little likelihood of success: There are no guarantees, however, that a resourceful adversary will not find ways to subvert existing techniques to their own benefit. Since cyber insecurity is likely to persist, a rethinking of the architecture of the Internet, and how it might evolve to become more secure, is warranted.

This chapter explores the interplay between Internet architecture and the ability of users, network operators and application service providers to adequately defend against threats posed by others on the Internet. It introduces the digital object (DO) architecture and suggests a way of integrating certain defined functionality into the Internet based on the use of digital objects. This approach is compatible with existing Internet capabilities and has the potential to substantially improve our ability to detect and deal with intentional hostile actions. It would also deal with actions that are simply accidental or naively misguided, but which may have serious consequences.

Today's Internet subsumes a wide range of networks, devices and other computational facilities, as well as diverse services, processes and applications. In order to protect against real and potential threats, technical capabilities are required to understand what is transpiring within the Internet and its various constituent components, and to take steps to deal with emergent

situations that may require action. For example, most laptop users have little or no idea what is transpiring on their computers, and no effective way to find out in real time. They may only know that something is not working properly, or that the machine is running more slowly than usual. At present, the Internet landscape is sufficiently complex that the myriad exchanges of bits over the Internet cannot easily be differentiated by intent or function. Certain architectural changes to the Internet, which primarily affect the way the Internet is used, can help in mitigating these situations. Specifically, the DO architecture can help remediate this situation.[1]

There are no guarantees that future threats, which require reconsideration of various architectural and design choices in the future, will not materialize; nor does use of the DO architecture guarantee that those who ignore or do not otherwise choose to take advantage of new architectural approaches will necessarily be harmed by that choice. At present, the Internet environment is tilted in favor of those with adverse motives, while the rest of the community must be on constant vigil to defend against harmful interference. However, over time, architectural changes become more pervasive. The assertion of this chapter is that the playing field will become more level in a way that provides architectural advantages for the defense of the Internet.

In the DO architecture, all system interactions involve the exchange of structured information in the form of digital objects, each of which has a unique identifier that can be resolved by a resolution system to state information about the object. Information, structured as a digital object, can be accessed and used by resources on the Internet based on its identifier, and is subject to any stated access controls or permissions associated with such objects. Even user commands, where invoked, can be converted into digital objects before being sent. This enables interoperability of the systems that embrace the protocol.

Digital Objects

A digital object consists of a data structure that is flexible, scalable and extensible. This data structure has a unique persistent identifier and may have one or more of the following:

- A set of type-value attributes that describe the object (one of which is the above mentioned object's identifier, which is mandatory).
- A set of named "data elements" that hold potentially large byte sequences (analogous perhaps to one or more data files).
- A set of type-value attributes for each of the data elements.

The elements of a digital object consist of "type-value pairs" that software at the destination and other locations can interpret for further processing. A protocol, known as the DO protocol, is responsible for managing the interactions between systems, services and other resources.[2] This protocol enables actions to be taken based on the use of identifiers. The actions to be taken, and the targets of those actions, are specified by identifiers, which relate to digital objects that prescribe the actions or enable access to the target information. This approach also enables verification of resources by clients/users, and clients/users by resources, since each client/user and resource also has at least one unique identifier. Indeed, a user may have multiple identifiers depending on the particular role the individual is playing at the moment (for instance, whether they are representing their employer or acting as an individual).

While many, if not most, interactions on the Internet are likely to be reasonable and legitimate, intrusions or hostile actions need to be flagged. Action must be taken to prevent damage, or other steps must be taken to quickly isolate matters. Even with the more structured view of the Internet provided by the DO architecture, the task is extremely challenging. Without such a view, the task is close to daunting, and would likely require semantic

interpretation of unstructured interactions, even if decrypted on the user's machine, that may be beyond the state of the art.

In the future, if arbitrary information arrives, the type of information will need to be understood from the structure of the information itself to enable further processing. Further, the environment into which the information arrives or is ultimately processed will require some degree of structuring, such as the structuring provided by the DO architecture, to determine with more specificity how best to deal with the information. In some cases, manual intervention may still be called for. In many other cases, however, automated processing may be possible based on interpretation of the structure of the actual information. For example, a medical reading sent by a remote wireless device might be understood from the structured information itself and placed in the user's medical record. Likewise, a remote financial transaction may be received and inserted automatically into a record of the user's daily transactions. Information collected in real time from remote sensors and appropriately identified can also be managed according to general rules and procedures adopted for such types of sensor information.

Overview of the Existing Internet Architecture

The existing Internet architecture was designed to enable the interconnection of multiple networks, devices and other computational facilities. Each potentially had a different design and performance, such that computers on different networks could communicate seamlessly and reliably with each other without having to know the location of the facilities, the intervening networks or how to actually route the information. More specifically, it enabled information in the form of packets of digital information to be communicated between computers without the need to first establish communication pathways between the computers. As a result, the Internet has become a standard means of communication worldwide, not only for traditional computer facilities, but also increasingly for digital representations of voice, video and sensor data managed by computers.[3]

At present, the Internet environment is tilted in favor of those with adverse motives, while the rest of the community must be on constant vigil to defend against harmful interference.

The Internet's creators based the existing architecture on two relatively simple notions. One was connecting networks with routers, which forward received packets by a process in which the routers act as relays with each step hopefully moving the packet closer to the eventual destination. The destination is specified by a globally unique identification known as an Internet protocol (IP) address that distinguishes the destination machine from all other destination machines on the Internet. The routers interpret the IP address to determine how best to route the packet. The process of communicating packets does not require the user to specify how to route the packets, which combination of networks to use, or even where the destination machine is located. Indeed, except for certain control information (such as the IP address) the contents of the packet may be encrypted. A dynamic routing protocol is used to adapt to changes in the underlying network components, such that if the packet can be routed to the eventual destination, it can be delivered in a timely fashion.

The second notion was the use of a host protocol, originally known only as the Transmission

Control Protocol (TCP), to enable the components to intercommunicate. TCP was later separated into two parts, one of which is IP, and the remaining part remained TCP. At the destination computer, TCP checks the validity of the arriving packets, discards duplicates that may have been generated along the way, reconfigures the data as appropriate and takes the necessary next steps in furthering the processing of the packets at the destination. In 1995, to clarify what the Internet actually was, the U.S. Federal Networking Council provided a definition of the Internet as a global information system that enables information resources of all kinds to intercommunicate by use of certain defined protocols (including IP) or their logical follow-ons and extensions.[4]

We note here that the overall objective of today's Internet is to ensure that global connectivity is achieved with low latency and reliable communication. While attacks on the network components of the Internet are possible, the Internet is far from completely defensible. Operators can take many types of precautions to ensure that traffic originating from users on their networks – and transit traffic from other networks – cannot directly cause actions within their networks (adverse or otherwise) other than to forward packets to their intended destination. However, although network operators can play a central role in helping to understand what is happening within their networks when adverse actions are reported or detected elsewhere, much of the concern still centers on vulnerabilities of the application service providers, their users and the underlying information systems they employ.

Vulnerabilities in Today's Internet

Various characteristics of the existing Internet make it especially vulnerable to harmful interference. One is the lack of overt security, which makes communications vulnerable to interference. Second is lack of identity management, which makes verification less secure than perhaps may be desired or necessary. Password protection is often used, but public key systems offer greater protection assuming the private keys are not communicated over the Internet. Passwords, which are communicated, may travel in the clear or be included in email messages (or perhaps accessible files), and can be used by anyone to access a password controlled system if they know the account name. Third is freedom of communication without prior arrangement that can include desirable or essential communication; however, this also enables undesirable communications, which may range from simply annoying to potentially harmful. There is a role for anonymous and non-pre-arranged communication in the Internet. But, at present, all communications are treated basically with equal significance, thus making it difficult to differentiate between those that are known and acceptable, versus those that are unknown and possibly undesirable. The key to addressing this issue lies with architectural changes in how information is managed in the Internet, including, in particular, in the devices and other computational facilities that provide the application services.

At present, all communications are treated basically with equal significance, thus making it difficult to differentiate between those that are known and acceptable, versus those that are unknown and possibly undesirable.

Much has been done to protect the various networks that comprise the communications portion of the Internet, and serious ongoing efforts exist to build ever more robust and reliable computational facilities. But, for the most part, the most severe

vulnerabilities in today's Internet exist in those applications – in operating systems and in other resources – that cannot adequately defend themselves. The extent of the threat possibility is still unfolding, but the earliest examples of intrusive action are by now well known. For example, spam is unwanted email that consumes communications capacity and can overwhelm user systems. But spam is increasingly being filtered out with the help of commercially available software designed to distinguish between spam and non-spam communications. Generally speaking, these software packages are not perfect, but they do reduce the nuisance significantly. Since most spammers rely on the dissemination of lots of similar traffic relatively indiscriminately, certain charging schemes could mitigate the spam traffic. However, most spam is not intended to cause damage, and some unwanted advertising might actually be of interest to some. In most cases, however, it represents an intrusion upon an unwilling recipient.

Other actions can actually cause damage in some form. Intrusions that penetrate user systems can collect private information, can harm or degrade the operation of the user's system and in extreme cases can render it unusable. These harmful actions are usually achieved by exploiting vulnerabilities in the operating system or in one or more applications that run on the machine. These actions result from incoming traffic generated by usually unknown sources that may have immediate effect, or may be the result of implants which arrived over the Internet much earlier. Indeed, one of the loopholes that many users are unaware of is that such intrusive software and implants may result from devices such as memory sticks that transmit them when inserted into the user's machine. Any individual whose memory stick has been compromised can (in principle) compromise any system to which it comes into contact. If you change the word "compromise" to "infect," the analogy with epidemiology becomes clear.

Finally, every network capability can be compromised by what are known as distributed denial-of-service attacks. These generally require coordinated actions by lots of machines on the Internet; and certain known types of attack can be mitigated or denied by the network operators who detect or are otherwise made aware of them. The first line of defense here must be the network operators.

How Best to Deal with These Vulnerabilities?

What can be done to deal with this situation going forward? Three assertions are made in this chapter, each of which is discussed further below. First, the DO architecture will help to achieve increased visibility and awareness into the possibility of actions that threaten systems that are part of the Internet. Second, a greater use of identity-based transactions on the Internet will ensure that – with the user's concurrence – the parties and perhaps devices and systems/resources involved in the transactions can be determined from the transactions, while still supporting privacy and allowing anonymous operations, if desired. Third, the use of an identifier-based mode of interaction with Internet resources may help to circumscribe the kinds of actions that can be taken and thus help to clarify the landscape whereby intrusions may occur. None of these steps, by themselves, will prevent clever individuals from seeking workarounds; but the architectural constraints can help to make the commission of unwanted actions more visible and harder to accomplish.

INCREASING VISIBILITY AND AWARENESS

When we drive a car, we have a general idea of what the car is and what is normal and abnormal behavior. We can determine if a tire is flat, or a headlight is out by direct inspection. By other clues we know that gas is required to power the engine and can sense when the tank may be empty, and can see the tank level from the gauges on the dashboard. In general, we have a degree of visibility into the current operation of our car. Similar statements can be made for many other things we come into contact with and depend on. No such statement

can be made about the computational facilities on which we depend or, for that matter, about the Internet itself.

Internet operators may know quite a bit about their networks and other computational facilities from information accessible in their control centers, and they are in a position to readily respond to many types of outages and disruptions. In general, they tend to have visibility into their networks and are aware of their current state and what may go wrong. While there will always be new situations they have not encountered before and situations in which they have no idea what is happening, their forensic staffs will undoubtedly be engaged to deal with these situations quickly. No such thing can be said if the situation is such that significant parts of the Internet are compromised. Remedial action by one network operator may only solve a piece of a more complex problem. While a global means of responding to a widespread threat is needed, this is largely a matter for policymakers from multiple nations to address in a political arena.

Users are generally in the worst position to respond to attacks and would have to rely on Internet defenses provided by others or contained in the software they use. Users typically rely on their computational facilities to carry out well-known tasks, and are usually much less knowledgeable than technical staff working for the organizations providing Internet services. For example, there is no serious equivalent of a user dashboard that portrays for the user the most important aspects of its computer in such a way that the user will know when something unwanted has happened, or makes it possible for the user to take action to repair the problem. Turning a machine off and then back on does nothing to deal with an implanted and potentially harmful virus, for example. Virus checking programs can help to prevent such unwanted intrusions, but, with today's operating systems and applications, clever perpetrators will easily find ways around commercial virus checkers and even hide the presence of harmful actors on a user's machine from subsequent detection.

Users should be able to inspect their computers with as much facility as they can inspect their cars. What might they like to know? Perhaps some would like to visualize the "actual" memory map of their computer to know what is stored in the various parts of memory – "actual" meaning what is really there, rather than what a program may be fooled to think is there. In addition, a user might like to know when traffic that makes it into or out of his or her machine is notable for some reason. A user might like to know about information flow that is unauthorized and to locate (and remove) programs that may be extracting information and shipping it elsewhere without permission or authorization. Further, users may want to access audit trails that provide information about how the unauthorized program was put on their machines, along with certain information that may already be available such as the time it was created on the machine.

With the DO architecture, a basis would be in place for better understanding what is transpiring within the Internet, thus yielding greater visibility into and awareness of potential threats. In this mode of operation, all operations are explicit and, with authorization, can be logged and diagnosed. In addition, the same can be done for entire sessions consisting of many transactions in series. Programs and users will have a smaller set of well defined primitives to invoke in their instrumentation; and presentations of results can be more succinctly prepared along with more detailed semantic interpretation.

While much of this area is still likely to be the subject of research and development for many years, some aspects can be addressed immediately. It remains to be seen, however, just how much information the average user will need or want in order to be a more informed Internet user in the future.

IDENTITY-BASED OPERATIONS

Critical information about users and their intended actions on the Internet today is largely unavailable

Users are generally in the worst position to respond to attacks and would have to rely on Internet defenses provided by others or contained in the software they use.

from or not visible from the information communicated. Further, such information may be encrypted and, thus, the intent would be purposely hidden while the information is in transit. The communications are from one machine with an IP address to another and otherwise consist of a flow of undifferentiated packets. Authorized users who wish to make use of remote machines are usually required to log into the remote machine and supply a password of some kind. Some systems allow anonymous usage (e.g. most search engines), but take steps (usually by severely limiting the number of possible actions) to ensure that users cannot harm their systems.

Let us postulate that every user has the ability to obtain one or more unique identifiers from one of potentially many bodies, each of which is known and trusted to authenticate assertions in digital form about individuals, including the mapping between such assertions and their unique identifiers. Efforts are underway in several quarters to formalize this mapping process, but such formal processes may not be required in many customary cases. The most convenient way to handle this is via individual actions involving parties that know and trust each other. For example, if a patient has an identifier he is comfortable providing to his doctor, the doctor can rely on that identifier for the purpose of providing information to that patient, since the patient would have authorized use of that identifier in the first place. If the identifier has associated with it a public/private key pair, and if the public key is accessible by use of the identifier, then a public key authentication can be invoked at any point the doctor or the doctor's information management system wishes to validate the patient. Similarly, if the patient contracts with a company to manage his or her health records, that company would have the obligation to make the connection between user and identifier.

An assertion about an individual that has a unique identifier acquired in connection with a desired task, process or service can be used to authenticate the user to a resource on the Internet. This provides a uniform way of validating the assertions. A similar process can be used to authenticate assertions about services, physical objects, organizations and other entities. When the service is remote, and the user learns of its identity from a third party, the user may elect to trust the third party (although this is not without its potential pitfalls) or to rely on bodies that maintain trusted information about such services.

However, users that do not wish to use their identifiers, or do not have identifiers, may still use Internet resources that permit such anonymous access. However, taking the route of anonymity may still allow services to be controlled in some situations where such control is deemed important or necessary. The main concern here is the provision of bogus identifiers by trust authorities or other entities. Using the term bogus does not mean that the identifiers are invalid, although that may be the case, but rather that the mapping of the identifier to assertions about the individual is not accurate or perhaps simply not known. These cases represent a kind of anonymity, but identifiers known to be linked to specific individuals may be unimportant in many cases, such as where payments are properly made or where accurate checking of identity is not critical. If problems were to arise here, one will know which identifiers were involved and perhaps who issued them in the first

place. Some regulation of the issuance of identifiers and the coupling of them to key pairs will be important, as is regulation of other trusted entities in society (such as banks).

Once a means of obtaining identifiers for individuals and organizations becomes routine, similar steps can be taken for Internet resources of all kinds. Systems and services can be given identifiers and users can validate them as easily as they can validate the users. Although accurate audits of information requested and disseminated can be enabled in this fashion, it also has the downside of enabling unauthorized accounts of such activity. In a free society, the balance of privacy versus security comes squarely into play here and requires careful examination from both regulatory and political perspectives.

Assuming all Internet information systems and other resources (including users, networks and devices, as well as the actual information or services being provided) have associated unique persistent identifiers, how would the operation of the Internet actually function in this context? How would informational resources be accessed in this manner? And why would it matter for Internet defense?

Circumscribing the Operations

If the main vulnerability of today's information systems comes from the operating systems and the applications that make use of them, an important first question is whether either or both of them can be avoided or if it is possible to otherwise constrain the vulnerabilities in some fashion. For some applications, the answer is clearly no, since they are essential to providing the desired user functionality. Most applications currently depend on underlying operating systems for many tasks such as storing files, scheduling multiple tasks and handling security and network functions. Vulnerabilities in the operating system pose direct threats to the application, yet many operating system functions will still be required. If some of the operating system functions are not really needed, however, perhaps that software can be simplified and made less vulnerable to attack.

Most of today's workstations, desktop and laptop computers are installed with a suite of application software, including office-related software for document preparation, spreadsheets and more. Downloads from trusted vendors are the norm, but subject to the vagaries of the user's system. Access to remote sites, such as those on the Web, are typically enabled via a Web browser, where each website complies with standard Web protocols and vulnerabilities in the browser protocols can have repercussions for users of the websites visited.

Reliance on structured information in the form of digital objects is another way to circumscribe the operations, since one knows both the nature of the operations to be performed and the targets of those operations. Digital objects, whether embodying what is traditionally viewed as "content" or actions to be taken on that content (perhaps in the form of executable code for which trust mechanisms can be invoked) can easily be incorporated within the DO architecture to enable a scalable and evolvable system going forward.

The largest growth in computational facilities has recently been with wireless devices, such as smartphones and tablets, where the devices may not be intended for use as general purpose computing platforms; and user desired functions that are not already installed on these devices are enabled by obtaining vetted computer programs (applications or "apps") usually written by others. Such apps can provide services of their own, or enable access to other resources on the Internet. Users typically activate these apps by touching the screen on their wireless device or taking an equivalent action. These apps can be customized by their providers to give a unique experience either using the device or in connection with a remote service or interaction. Thus, suppliers of such apps are usually not constrained by the technology to

any single set of application protocols or means of presentation, but those made available with the user's device are often the most convenient to use. By this measure, the Web, along with the Web browser, is but one very pervasive app.

Apps, in general, may not require many services typically provided by an operating system. In this chapter, it is assumed that the operating system may be viewed as a mini-version of a combined traditional operating system with a high-level programming language, which we call "MyOp" for short. MyOp is assumed to provide a well known programming language execution environment, network access, maintenance of address books and/or mailing lists, the ability to select and schedule resources for execution and the ability to execute public/private key encryption and decryption. It is assumed that usual file and folder operations are replaced by use of a special purpose app that provides repository functions and uses either internal storage (if necessary), external storage (if available) and possibly both under certain conditions. Synchronization functions are not discussed here, but these could be embedded in MyOp or combined in the repository app.

MyOp is assumed not to be programmable by third party computer programs, and since apps cannot directly interact with other apps except by communicating with them via information structured as digital objects, this should limit the vulnerability from external threats to manifest themselves through unknown installed "hooks." It remains to be seen whether it will be possible to inhibit apps from permitting the execution of third party digital objects that are executable programs. If not, the use of specialized sentinel programs called "bastion objects" that cordon off the range of operations of such apps may be required. If a user can be aware of all the downloaded apps on his device, he can be made aware if an unwanted app were somehow to arrive. In any event, since he would have taken no action to cause it to be downloaded (of which he was aware), either his system would detect it to be unwanted and take appropriate action or, somehow, his system would have had to be fooled into making such a request (or getting his system to think such a request had been made). All this is to explain how the discourse of dealing with threats and defense against such threats would shift from a wide unknown range of possibilities to a situation in which various types of attack scenarios can be better described and thus dealt with both before, during and after the fact.

No other actions are allowed by any app relative to MyOp, and further no app is permitted to interact with any other app except by passing identified information, referred to here as digital objects. So, temporary or permanent storage of digital objects takes place via the internal repository app or by passing the information to an external repository. Digital objects are constructed by the repository app, or by APIs (application programming interfaces) that make use of it, according to a meta-level standard and parsable structure understandable by apps throughout the Internet; a unique persistent identifier is also associated with each such digital object. Thus, all arriving and departing information is in the form of digital objects, and internally generated information that does not leave the local computational environment is also stored as one or more digital objects.

Information in the form of digital objects flowing over the component networks of the Internet can thus be individually identified along with all incoming and outgoing information from any device or other computational facility. Although there is no requirement that any part of this information, including its associated identifier, be made visible in the network, users may wish to make the identifier part of a given digital object visible for any of several reasons. One is that the provenance of the information can be made available when the information becomes available. Another is that users can require that references to responses from their systems include the identifier of each digital object being responded to for cross-correlation or validation on receipt. Coupled with

timestamps and use of public key encryption, this approach can also be used to validate individual steps in a series of transactions or other operations taking place during a single session.

Large server farms will have very different needs than an individual user's computational devices, but their level of expertise can be expected to be much higher as well. No matter what the level of expertise, however, if such server farms require more sophisticated operating systems and related services to support distributed computing (sometimes referred to as "cloud computing") within and among the servers in the farm, care will have to be taken to identify, isolate and hopefully remove latent system vulnerabilities. Internet-based server farms, particularly if they store large amounts of data, provide specific targets for potential attackers. Thus, a combination of local storage and remote storage might provide a reliable approach in the event of sabotage or denial-of-service. Normally, one might rely on remote storage for day-to-day operations and only use the then-current local storage choice in those cases for which the remote storage is unavailable. If the remote storage is disabled or destroyed, or cannot otherwise be brought back up for days, weeks or months (or longer), a user can temporarily resort to the user's local storage capability.

It is assumed these server farms can be operated both reliably and securely. However, users may wish to store their digital objects in encrypted form, with the keys kept separate from the remote storage site. In this case, operations with the remote storage site will likely be of the warehousing variety with entire digital objects being passed back and forth. When encryption is not required or is not invoked, operations with the remote storage can be more fine-grained, and specific elements of the digital object may be accessed directly or after performing one or more remote operations without the need to retrieve the entire object. Recent developments have shown that remote interactions with encrypted objects are also possible in certain cases, but this aspect is not explored further in this chapter. In cases of very large objects, which would consume bandwidth and take time to transport, the ability to access directly specified parts of the object would have obvious appeal.

In each of these cases, the potential number of digital objects can be quite large and users cannot, and indeed will not, be able to remember their identifiers, even if they can recall attributes of the digital objects to which they were assigned. Software known as registries serves the purpose of allowing users to register such objects, presumably automatically in most cases and manually (if desired) in others. These registries can be installed as separate apps on the user devices, or provided by server farms over the Internet. In both cases, the registry metadata will be produced either manually by the user or automatically at the time the original digital object is created. Indeed, the user should be able to annotate such metadata and have it apply to the metadata pertaining to a specified range of digital objects.

If a user's device is lost, he may lose the apps that were available on it, but some vendor implementations should permit the user to access such programs over the Internet at no additional cost and inhibit the operation of that app on the lost device. At a minimum, this capability would seem to require each such computational device to have its own unique identifier, and perhaps be able to hear about such loss via MyOp; however, other means of disabling such apps are also possible.

In this model, the role of IP addresses would remain unchanged, along with the role of routers and networks that interpret them. In addition, those components would have the added advantage of using the digital object identifiers to meet stated objectives as well. The DO architecture can thus be integrated into the existing Internet as well as working in other communication systems. To clarify this point, in a proposed modification to the 1995 Federal Networking Council definition,

the Corporation for National Research Initiatives (CNRI) recommended adding the words "or integrated with" to the section that talked about applications layered on the underlying protocols.[5]

In an architectural environment where all accesses to systems, services and other resources are managed using identifiers for each such resource, and all information is structured in the form of digital objects, the task of Internet defense is altered in several fundamental ways. When operations in the Internet can be made more structured, one no longer has to be on the lookout for bit patterns whose purpose and intent cannot easily be determined. If, as a result, most actions consist of a more limited set of types of basic operations (which the author refers to as "meta-level operations" to reflect the fact that they indirectly reference the actions to be taken and their targets), it may be possible to develop protective steps that are more effective. This is definitely not the case today. If the digital object architecture were integrated within the Internet, its operations and targets would be separately identifiable so that, from these identifiers, the digital objects that were involved could be determined from the metadata, and the users could (if they choose) retain all the associated digital objects for later analysis (if desired). Many other properties of the communication could also be acquired, such as timing data for each digital object (e.g., creation, dispatch and arrival) should that be of interest. This is particularly important in connection with emerging Internet capabilities that relate information about "things" to other information in the Internet.

A user who is well aware of what is happening on his device will ordinarily be in a position to take manual action if necessary. First, he has to be paying attention, which may not always be the case. Second, an attack may have significant negative impact within seconds, or even microseconds. Thus, the ability of a system to respond in kind would seem to be essential. Efforts to develop cognitive systems that understand their environment, their own capabilities and modes of behavior, and threats to their operation have been undertaken in the past; but the task has remained daunting by virtue of the many degrees of freedom posed by the general problem. In other words, there are just too many things to have to know about, look for and react to. With the digital object architecture, the number of possibilities is greatly reduced and, thus, the likelihood of success is potentially much higher. An environment where threats could be internalized within a system, and where the system can defend itself with mobile programs specifically tasked and authorized to take actions against fast moving attacks, would provide an immediate benefit to the user by defusing the attack in real time. It could also serve to provide data for a post-mortem report on the attack.

As a matter of policy, it would be useful if users can work with the involved carriers or other relevant service providers when such problems arise to determine what happened. This can be helpful in determining what networks, proxy servers or other related infrastructure or resources may have been compromised, and how best to thwart any such ongoing incidents. This would potentially have the effect of enabling legitimate backpressure or other corrective action wherever required in the Internet.

Conclusion

The digital object architecture would impact the nature of many Internet activities by making them more explicit and, thus, potentially more defensible against attack. It would help to support an informed discourse about implementation of effective Internet defense strategies that are difficult to achieve today. The continuing transition to the DO architecture is an incremental process that may take years to complete. In the meantime, considerable progress could be achieved (especially for users) in understanding what is transpiring on the Internet (including on their machines and devices), and working with Internet service providers to ensure that undesirable events can be more easily diagnosed and prevented, or at least detected and hopefully defused before they cause substantial damage.

ENDNOTES

1. Peter J. Denning and Robert E. Kahn, "The Long Quest for Universal Information Access," *Communication of the ACM*, Vol. 53, Issue 12 (December 2010): 34-36, http://dx.doi.org/10.1145/1859204.1859218. See also Corporation for National Research Initiatives, "A Brief Summary of the Digital Object Architecture" (1 June 2010), http://hdl.handle.net/4263537/5041.

2. Sean Reilly, "Digital Object Protocol Specification," Corporation for National Research Initiatives (12 November 2009), http://dorepository.org/documentation/Protocol_Specification.pdf.

3. See Robert E. Kahn and Vinton G. Cerf, "What is the Internet (And What Makes It Work)," Corporation for National Research Initiatives (December 1999), http://www.cnri.reston.va.us/what_is_internet.html; and Robert E. Kahn, "The Architectural Evolution of the Internet" (17 November 2010), http://hdl.handle.net/4263537/5044.

4. U.S. National Coordination Office for Networking and Information Technology Research and Development, "FNC Resolution: Definition of Internet" (24 October 1995), http://www.nitrd.gov/fnc/Internet_res.html.

5. Patrice A. Lyons, "The End-End Principle and the Definition of Internet," Corporation for National Research Initiatives (10 November 2004), http://www.wgig.org/docs/CNRInovember.pdf.