Security, privacy and trust issues raised by the Personal Server Concept

Paper for the
Workshop on Security and Privacy in Pervasive Computing
at Pervasive 2004

John Light, Intel Research

Abstract

This paper is a survey of user risks associated with the Personal Server concept. The Personal Server concept is used as a surrogate for future mobile devices. The risks are threats involving security, privacy and trust. An overview of the concept is provided, followed by descriptions of three usage models: mobile storage, application server, and beacon receiver. Each usage model description includes a discussion of risks that result from that usage. No solutions are provided.

The Personal Server concept

Among other ends, Pervasive Computing deconstructs the User Interface which has dominated computing for the last two decades. Since the Personal Computer (from Apple and IBM) arrived in the early ‘80s, User Interface has consisted of a human sitting upright in front of a vertical display surface wielding a keyboard and pointing device on a horizontal surface. This paradigm is unchallenged for “real” computers, but the advent of Personal Digital Assistants and especially cell phones has challenged it in the larger arena.

The Personal Server project [5,6] explores an extreme alternative approach to this paradigm by asking “What if your computer had no standard user interface?” How would that change what our computers consists of and how we use them? How would the world have to change in order to accommodate us? How would that change how we feel about computing? How would it change the impact computers have on our lives?

To explore these questions we created a mobile device with considerable processing power, storage, battery capacity and communication capability but no display or input device. It is a fully capable computer without an inherent user interface. We don’t expect to see a product built this way, but we hope that what we learn can be applied to building better mobile computing devices of all sorts.

The Personal Server prototype consists of an Intel PX255 processor, which includes Intel XScale® technology, two Compact Flash slots for memory expansion, a Zeevo Bluetooth radio, and a battery capable of running the device for about a day. The prototype is being manufactured and sold by Crossbow Technologies for the benefit of researchers in many disciplines who want a compact, highly capable mobile computing platform. An open source Linux distribution is available on SourceForge to support it. Compact Flash cards with capacities of up to 4 gigabytes are currently being sold, and larger ones have been announced.

The Personal Server is analogous to a personal version of the back-end servers that provide file, web, database and application services to desktop computers. Just as the Personal Computer took the mainframe computer out of the back room two decades ago, and the notebook PC took the Personal Computer out of the office, and the PDA took the PC onto the street, the Personal Server takes the back-end server out of the back room and puts it in the pocket or purse. An important implication of this analogy is that while PCs of all sorts are often turned on and off, servers tend to be “always on”, providing services even when the user is not directly engaged. The Personal Server is designed to run all day in the user’s pocket, and this is a characteristic it shares with the cell phone.

Capabilities of a Personal Server may eventually be included in some other form of mobile device, such as a Personal Digital Assistant or cell phone, since its physical components are very similar to both.

The immediate questions posed by the Personal Server concept are:

- What computing needs can such a device satisfy?
- What personal needs can such a device fulfill?
- How does one interact with such a device?
- Can interaction with such a device be effective and satisfying?
- Can interaction with such a device be safe?

This paper explores the issues related to last question using our learnings from the other questions.

Summary of security and privacy issues

Because the Personal Server explores an extreme computing model, it raises unique issues of security, privacy and trust in addition to those present in any mobile device. We expect aspects of the Personal Server to make their way into mainstream products in the future, and the Personal Server project provides a relatively clear view of what those issues may be.

Any mobile device raises concerns about security (“Can someone modify or destroy my data?”), privacy (“Can someone read my data?”), and trust (“Can I count on my data being available when I need it?”). The way these issues manifest themselves depends on the nature of the device, the nature of its use, and the expectations of its user.

The Personal Server concept expands on those issues because of its lack of display and dependence on a wireless connection to the world. For any computer system, the most severe threats involve external communication, and **all** of the Personal Server’s operations involve interaction with external sources. Moreover, the Personal Server concept proposes new primary modes of external interaction such as annexing external

User Interaction devices and listening to Information Beacons. Annexation raises new questions for secure authentication, and listening to beacons raises new issues of privacy.

This paper summarizes the security, privacy and trust issues uncovered by the Personal Server project. We will not explore issues that are common to all mobile devices, concentrating on those that are unique to Personal Server concept. We hope that this exposition of issues will add to the overall picture [4] of what we need to do to make the Pervasive Computing environment safe.

Generic risks

Any mobile device carries risks involving security, privacy and trust. Solutions to eliminating or mitigating such risks are an on-going effort by the mobile computing community. The Personal Server project assumes that those efforts will be successful and expects to benefit from them. We will survey them quickly to provide a more complete picture of the issues.

At one extreme of the mobile device playing field are the smart card and USB Flash storage device, sometimes called a USB dongle. Both have a primary purpose of carrying information safely from one place to another. Both are implemented with storage and a processor sufficient to interface them to other computing devices, and that is their primary purpose. In one case the storage and device size are very small (smart card), and in the other case (USB dongle) the storage capacity can be quite large in a package not much bigger. The biggest difference is that the smart card is designed to only talk with trusted readers while a USB dongle can connect with nearly any computer.

At another extreme is the notebook computer. Some are barely mobile, and they typically include large amounts of storage. Most have many I/O mechanisms, but I/O other than the keyboard, display and pointer is usually of secondary importance. The primary purpose of most notebook computers is as a more or less complete, self-contained computing environment. A notebook computer may be just as vulnerable to risks of security, privacy and trust, but many of those risks can be mitigated by working without connection to the external world until a safe venue is attained.

Most mobile devices fit within those extreme, but they all share some common concerns.

- How likely is the device to be stolen?
- How likely is the device to be lost?
- If it is lost or stolen, what is the likelihood that its contents will be stolen?
- If it is lost or stolen, how quickly and easily can it be replaced?
- If it is lost or stolen, how much information and work will I lose?
- Can its contents be stolen during normal usage?
- How susceptible is my interaction with the device to being observed?
- Can someone introduce a malign agent into the device?

Some of these concerns are bigger problems for some devices than others. The likelihood of being stolen is a complex function of perceived value versus perceived risk on the part of a potential thief. A device that is often put down on surfaces is more likely to be stolen or lost. Moreover, the availability of effective (and used) security and privacy technologies can make the loss of data less of a problem. The availability (and use) of backup or synchronization services can mitigate the replacement problem.

The Personal Server and other devices with Personal Server capabilities are vulnerable to these same risks. We expect products with these capabilities will use the best known practices to deal with these and other generic risks. The rest of the paper discusses risks that are introduced or emphasized by the Personal Server concept, which we will refer to as *incremental risks*.

Mobile storage issues

The earliest usage models explored on the Personal Server were its use as a file and web server. These are traditional uses of a traditional back-room server, and the Personal Server's wireless file server capability is an obvious extension of the current popularity of USB dongles. Portable storage devices have always held an important place in personal computing, and USB dongles have largely inherited the place once occupied by floppy disks.

The obvious difference in this use with devices such as the USB dongle is the wireless connection provided by the Bluetooth radio. Instead of reaching into your pocket for a USB dongle, fumbling with your computer to plug it in, and trying to remember to take it with you when you leave, you can use the Personal Server while it stays untouched in your pocket. This simplicity of use comes at the cost of some implementation complexity and incremental risks.

One class of incremental risks for the Personal Server involves the nature of wireless connections. When your USB dongle connects to a computer, it is typically obvious what connection has been made: the physicality of the connector ensures the integrity of the connection. A wireless connection, on the other hand, can be ambiguous. How do I know what connection I've made, and how do I know there is not a "man-in-the-middle"? There are no natural physical artifacts to answer those questions.

Any storage device must be able to reliably hold data. A mobile storage device must deal with physical threats to the device, e.g., theft, dropping, losing, etc., which are normally dealt with by some form of synchronization or backup. Furthermore, the normal usage of such a device exposes it to hosts outside of the user's direct control, e.g., a friend's or customer's notebook computer, etc., which exposes it to intentional or unintentional data loss. Some storage devices include a physical switch to write-protect the contents, but such switches are hard to use, so small that few people even know they are there, and unlikely to be used at critical times. They also provide only binary control: if anything is to be written, then all protection goes away.

A mobile storage device should be able to hold data securely. Hard drives typically depend on the physical security of their location to provide data security, but a mobile storage device is more likely to fall into the hands of someone who wants to steal the contents, through either theft or loss of the device. Furthermore, the normal usage of the device exposes its contents to theft whenever it is connected with a host not directly controlled by the storage device owner. This is true whether the host is operated by the user (a rented computer) or not (a customer computer). Most current devices expose all their contents whenever they are plugged in, and the few with authentication methods expose all their contents after authentication succeeds. Ideally, only the data relevant to a transaction would be accessible at any one time.

A mobile storage device must provide reasonable access to its held data. The word "reasonable" refers to a tradeoff between the user's risk and effort. Security often deals with such tradeoffs, but the need to include untrusted hosts in the security equation makes solutions more difficult. For example, common security methods such as typed passwords are less effective in the common usage model since they expose the passwords themselves to theft. This can lead to more complicated security measures, which may discourage using either the device or the security measures. It is not sufficient to prove that a procedure is secure unless you can also prove that people will use it. This problem encourages the development of alternative authentication methods.

A mobile storage device can act as a vector for worms, viruses and other forms of malware. Because it promiscuously connects to multiple devices and connects quite directly (typically as a mapped file system), it is an ideal vector for malware. All such devices are currently vulnerable to existing viruses, and we expect malware to be written specifically for mobile storage devices as the use of such devices proliferates. Since the current crop of mobile storage devices are seen as big floppy disks, this problem is being treated as a host issue, but it is not practical to scan all the contents of a multi-gigabyte storage device every time it is plugged into a host. The device itself must be involved in supporting the protection process, and the host must be able to trust that involvement.

The Personal Server project has explored solutions for some of these problems, using the device's processing power to counter its vulnerability. For example, we have considered structured availability of data, new forms of authentication [2], and access journaling. The Personal Server can also present its contents in the form of a Web site, which reduces some threats to the Personal Server but not the host. Discussion of these solutions is not within the scope of this paper.

Application server issues

The processor of the Personal Server allows it to act as an application server. In this case the data for an application is stored on the Personal Server, and a program that implements the application runs there as well. In some cases the application can run with little or no user interface, but in others a user interface is needed. If the Personal Server capability is embedded in a device with a display screen, that screen might be used for the application.

Some applications require a bigger screen than a mobile device can reasonable provide, and some applications involve collaborative use with co-located individuals. In those cases, an external screen might be used with a mobile device. Desktop computer users have had remote access to their machines for years, and we believe this capability may become common with mobile devices as well. Thus, this problem is not limited to the Personal Server model.

The model here is that someone with a mobile device (e.g., a Personal Server) would walk up to a public display, take some action on that public display, and create an interaction session on that display with an application running in the mobile device. For the duration of the session, the user would use the affordances of the public display to interact with the application and the results would be shown on the public display.

Known as *annexation* [3], this use of an external interaction device can provide a larger or shared screen when needed. Several relevant problems arise from annexation.

- How do you know which display you are annexing? This may seem obvious, but if you annex an interaction device that someone else controls, they might steal or destroy your information before you even know there is a problem.
- How do you know the interaction device isn't recording your session? There are lots of nefarious uses for a session recording.
- How do you authenticate yourself to your mobile device without exposing passwords? This problem is common with the previous section on mobile storage devices.
- How do you know that your interaction session is controlling your mobile device? An observer might be able to simulate your typical session with another device (after observing a previous session) well enough for you to be fooled into typing sensitive information into it.
- How do you know there is not a man-in-the-middle passing your interaction through until you have authenticated yourself? The man-in-the-middle may then either steal information or take control.
- How do look at information on a public display without displaying more than you want?

The last question is really a whole class of questions about how we deal with information in settings that are not entirely private. The advent of Pervasive Computing and the transformation of the office are combining to make our work places more communal or public and less private. Our databases and web sites are often not organized according to sensitivity of information so accessing one piece of information often exposes other pieces that shouldn't be exposed. In the privacy of an office this is usually acceptable, but in many other places we would like to work, it is not.

Since the Personal Server is "always on", it can run applications that might operate independently of user involvement. Such software *agents* can recognize context, respond to events, monitor activity, and notify the user, according to the expressed preferences of the device owner. The agent may operate based on external events, and the veracity of those events may be doubtful if the device is under attack. Such agents should be designed to deal with uncertain and false events. More importantly, an agent may be empowered to act externally to the device on the user's behalf, and these actions may need to be performed without user involvement with untrusted external devices. This creates new security challenges.

Information Beacon issues

The wireless capability and "always on" behavior of the Personal Server allows it to act as a receiver for wireless *information beacons*. Information beacons are small wireless transmitters with a relatively small (~10 meter) broadcast radius. They are inexpensive (<US$25), so anyone (store owner, individual, government, etc.) can place them wherever people walk by carrying appropriate receivers. A short repetitive message (~10K bytes) can be received by any receiver as it passes a beacon.

The combination of information beacons and receivers create a new form of location-aware computing, previously described in a workshop at UbiComp 2003 [1]. It requires no central authority for registration, location mapping, or content handling. Instead, the information passes directly from its source (who owns the information beacon) to its destination (who owns the receiver). The Personal Server can run software agents that process the incoming beacon messages and act on or archive them without direct user involvement.

Any form of location-aware computing raises issues of privacy and trust. We believe the use of information beacons raises fewer such issues than other forms of location-aware computing since it doesn't involve third parties such as cellular vendors or location-database web sites and it doesn't require traceable radio activity on an ongoing basis. Comparing the use of information beacons with other forms of location-aware computing is not in the scope of this paper. We will summarize the privacy and trust issues of this new approach.

Information beacons offer information and services to passing receivers. The information might be as simple as a store description, or it might include a full menu for a restaurant or a coupon for a clothing store. It could offer to sell something to the user, and the transaction might be able to take place immediately. Previous forms of location-aware computing have concentrated on immediate notification of "interesting" events because of the high cost of maintaining and processing significant state in a centralized resource for each user. We believe the cost of handling state can be much lower in a distributed approach. The new approach concentrates on building a personalized location database for the user, providing a useful source of context and state computations and reducing the need for interruptions commonly seen in other approaches. An agent running in the receiver might interrupt the user, but it would be based on considerably more context than is available to some other approaches.

Three classes of privacy and trust issues arise with the new approach. One class involves external tracking. If a user is communicating continuously with a series of information sources as she passes through an environment, software with a global view of the information sources could track her location and path. This is similar to the concern that the cellular network can track you while you carry a cell phone. This problem can be mitigated by avoiding use of a traceable identifier in communications with the information beacons. The problem can be eliminated entirely if the transmissions are entirely unidirectional. That is, if the receiver doesn't have to send any radio message in order to receive the beacon information, then there is essentially no way for the receiver to be tracked.

Another class of issues involves self tracking. As the receiver collects information from beacons, it likely creates a time stamped record of locations in its persistent storage. This record can be a major source of value for this approach to location-aware computing, but it can also be a risk in the case that a receiver is lost, stolen or subpoenaed. To mitigate this risk, the user should have full and nuanced control over both the collection and retention of such data. By "nuanced" we mean that the user should be able to have detailed control over various aspects of the data collection and retention, not just the ability to enable and disable.

The third class of issues involves user preferences. The agent that responds to beacon messages must be configured to behave as the user wishes. These preferences form a personal database that may be quite sensitive, depending on its contents. A user may want a mobile agent to work in the more personal parts of her life, and the preferences expressed to that agent may be especially sensitive. The point is that metadata may create as much of an incremental risk as data.

The use of information beacons is an exemplar of the class of applications that can be built on an "always on" platform. Any such program that interacts with the outside world via radio, infrared, RFID, etc., is likely to have similar issues with privacy and trust. As with location-aware computing there are often multiple approaches to architecting the system. The architecture that is easiest or most obvious (or appears to have the most revenue potential) may not be the one that offers privacy and trust.

Summary

Because the Personal Server defines a new computing model and new usage models, it exposes new risks to security, privacy and trust. Whether the Personal Server as presented here ever becomes a product is not important, but it is clear to us that various capabilities of the concept will become part of other mobile devices. The Personal Server project provides an opportunity for us to identify these risks at an early stage and provide solutions before they are needed. This paper describes what has been learned so far about risks facing any mobile device that incorporates aspects of the Personal Server concept.

Acknowledgements

This paper represents work of the Personal Server project of Intel Research and includes contributions by all project members, including Trevor Pering, Murali Sundar, and Roy Want.

Bibliography

1. Light, J., Pattison, E., Pering, T., Sundar, M., Want, R., Fully Distributed Location-Aware Computing, UbiComp 2003 workshop, http://research.microsoft.com/workshops/UbiLoc03/2003WorkshopOnLocationAwareComputing.pdf
2. Pering, T., Sundar, M., Light, J., and Want, R. (2003) Photographic Authentication through Untrusted Terminals. Proceedings of IEEE Pervasive Computing, 2(1), 30-36.
3. Pierce, J. S., Mahaney, H. E., Abowd, G. D. Opportunistic Annexing for Handheld Devices: Opportunities and Challenges. Proceedings of the Human-Computer Interaction Consortium 2004 (Feb 4-8, Fraser, CO), 2004.
4. Schneier, B., Secrets and Lies: Digital Security in a Networked World. Wiley, 2000.
5. Want, R., Pering, T., Danneels, G., Kumar, M., Sundar, M., Light, J., The Personal Server: Changing the Way We Think about Ubiquitous Computing. Proceedings of UbiComp 2002: 194-209
6. Want, R., Pering, T., New Horizons for Mobile Computing. Proceeding of PerCom 2003.