2008-01-01

Ubiquitous Desktops with Multi-factor Authentication

Paul Doyle

Technological University Dublin, paul.doyle@tudublin.ie

Recommended Citation

Doyle, P. (2008). Ubiquitous Desktops with Multi-factor Authentication. *Digital Information Management, (ICDIM 2008): Third International Conference*, London, 13-16 Nov., pg. 198-203. doi:10.1109/ICDIM.2008.4746797

Ubiquitous Desktops with Multi-factor Authentication

Paul Doyle, Mark Deegan, Ciaran O'Driscoll, Michael Gleeson, Brian Gillespie
School of Computing, Dublin Institute of Technology, Kevin Street, Dublin 8, Ireland
{paul.doyle, mark.deegan, ciaran.odriscoll, michael.gleeson}@comp.dit.ie, *brian.gillespie@simtone.net*

Abstract

There is no single desktop which meets the growing user requirement for diverse services and solutions. Desktop services such as operating system specific productivity tools, laboratory teaching environments, product testing and program compilation typically require substantially different desktop environments. While virtualization addresses the consolidation of multiple operating system environments onto a single piece of hardware, the issue of consolidating these multiple services into a single desktop for users must also be addressed. The need has arisen for a secure, mobile and integrated desktop which provides simple access to multiple desktop services. This paper proposes a ubiquitous desktop solution addressing these issues through a combination of virtualization, multi-factor authentication, single sign-on and thin client technology.

1. Introduction

As the demand for desktop based services increases, the traditional single purpose desktop cannot meet the growing and diverse requirements of users. Whether it is a specific application, operating system, network, or configuration, a single desktop cannot be expected to service all requirements. For example, within academia the same desktop cannot be used for teaching courses such as networking, system security, operating systems, and programming in multiple operating system environments. Within industry, a single desktop is not suitable for performing multiple tasks such as product integration, unit testing, system demonstration, and performance testing. The number of physical machines owned, administered and operated per user is increasing and while there is a growth in the use of virtualization software to consolidate the hardware requirements, virtualization alone does not fully address the complexity associated with integrating and managing multiple desktops. The ubiquitous desktop design discussed in this paper not only addresses the issue of complexity but also discusses multi-factor authentication and service mobility.

A ubiquitous desktop is a logical association of multiple desktop computing instances (which we will refer to as services) accessible from any number of stateless thin clients with all services authenticated using a single sign-on multi-factor security system. The centralized authentication process, desktop mobility, and grouping of services all contribute to a focused and simplified user experience. The ubiquitous desktop design takes us a step closer to a ubiquitous computing environment [1]. To provide context for the proposed design, a review of state-of-the-art desktop implementations in academia and industry is performed in Section 2. Section 3 reviews requirements and relevant technologies while Section 4 describes in detail the proposed new design.

2. Related Work

Within industry and academia there have been significant advances in the delivery of desktop environments to users, mainly through the use of virtualization and thin clients. These solutions are limited to providing single desktop experiences to users while the ubiquitous desktop solution delivers multiple desktops to users using a single sign-on process.

2.1. Academia

Increasingly institutes are redesigning their internal infrastructure to provide greater levels of service and productivity to both staff and students. Diverse desktop environments are often required within computer laboratories for the purpose of teaching, practical project work and demonstrations. Operating system installations, kernel debugging, building network services and testing security services are among some of the critical activities usually requiring administrator privileges. Desktop configurations must not interfere with the institute services (for example, creating alternative DHCP servers within the campus network while teaching a course on network configuration) and must ensure that systems are available to the all students when the laboratory is completed. There is no single desktop service which fits all requirements and institutes are increasingly turning to some form of virtualization to address this issue. Some recent work in this area is reviewed below.

Nieh describes a VMware [2] Workstation based virtual laboratories which provides VNC access to virtual machines running on laboratory machines, allowing students access their laboratories from around the world [3]. Border describes a similar VMware solution based on RDP rather than VNC [4], while Hu offers a Xen based solution [5]. Adams extends the virtual laboratory through the use of central NFS storage for virtual machines [6] and Villanueva uses a VMware GSX server to run virtual machines on a central server which can be accessed remotely via a browser allowing the laboratory machines to remain virtually stateless [7]. Finally, Miller discusses virtualization of desktops using VMware and RDP [8]. One of the limiting factors of each of these solutions is that only one desktop environment is accessed at a time. There is no provision for integrating multiple desktop views to the user at the same time.

2.3. Industry

The advantage of thin clients and virtualization are well recognized within industry with major initiatives ongoing. There is clear evidence of a trend towards desktop virtualization. In most cases, however, the focus is on single desktop services delivered to the user. Some of the more recent trends are described below.

SunRay [9], the thin client solution from Sun Microsystems, provides a client and server solution with token based smartcards allowing users to move from one thin client device to another and have their "session" follow them by tracking the token ID on the smartcard. SunRay is widely deployed within Sun and has a growing commercial customer base. IBM recently announced its "Ready-to-Use Cloud computing initiative" where IT services are offered via the internet. It utilizes Xen and PowerVM virtualized Linux operating system images, bringing virtualization infrastructure beyond the local network and into the internet (Cloud) [10]. Amazon's Elastic Compute Cloud (EC2) offers user owned and run virtual machines accessed and controlled from anywhere on the internet [11].

3. Ubiquitous Desktop Requirements

The requirements for a ubiquitous desktop are listed below. While the solutions reviewed in the previous section incorporate some of these requirements, it is the combination of these requirements that is central to this paper. These points identify the key characteristics essential to the proposed design.

a. Multiple services delivered to a single user
b. Multi-factor authentication
c. Single sign-on event
d. Mobility of the desktop between multiple access devices

3.1. Multiple services delivered to a single user

For the purpose of this paper a service is defined as a preconfigured Virtual Machine Guest Operating System. From a single ubiquitous desktop, multiple services are available to the user in a simple and consistent manner (Figure 1). Examples of services (all of which can be on a variety of operating system flavors such as Windows/Unix/Linux) are shown below.

a. Productivity Services (Mail/Internet Browsing/Word Processing)

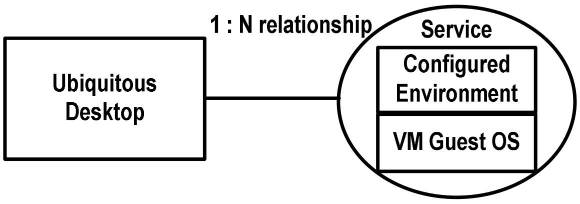

Figure1. Multiple services are associated with a single desktop

b. Programming Services (Preconfigured compilers on specific operating systems)
c. Testing Services (Pre-release software installations)
d. Networking Teaching Services (e.g.: isolated environments for teaching DNS/WINS/DHCP)

3.2. Multi-factor authentication

Multi-factor authentication requires the use of more than one authentication factor. Typical authentication factors include the following: what the user knows (usually a username and password), what the user has (e.g.: SmartCard, USB Key), where the user is (location based identification such as GPS and terminal identification) and who the user is (biometrics such as retinal scanning or finger printing) [12]. Within the banking industry for example, the dispensing of cash from an ATM requires triple factor authentication. Firstly, the ATM (where we are) determines if the user is allowed to access cash from that specific location. Secondly, the ATM card (what we have) must contain valid account data. Finally, the PIN (what we know) must correspond to the encrypted PIN associated with the ATM card. Many existing desktop environments typically use single-factor authentication (username and password). However, as the number of services accessible from a single sign-on authentication event increase, the use of multi-factor authentication should be seen as a requirement. With multiple services now accessible from a single sign-on the stakes are higher in the event of an unwanted intrusion.

3.3. Single sign-on events

A single sign-on event occurs where the authentication process authorises access to multiple services. If there are three services associated with a user, authentication is not required per service. Once authenticated, any service associated with the user will be accessed without additional username/ password or other authentication requests being required as shown in Figure 2.

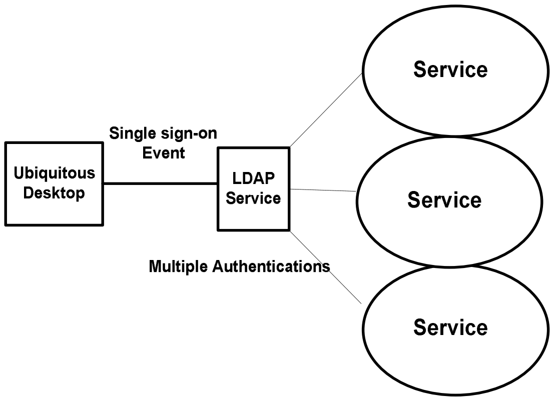

Figure 2. Single sign-on for multiple services

3.4. Mobility of the desktop

The ubiquitous desktop can move between different access devices (thin clients) to access services and is not fixed to any specific device (Figure 3). Services may remain active when the user is not connected. The access device is a stateless machine providing visual and audio rendering of services. None of the services will execute on the thin client which has a primary function of rendering the service.

4. Ubiquitous Desktop Design

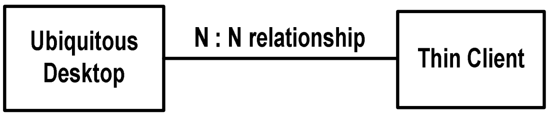

Figure 3. Mobile services where any ubiquitous desktop can be accessed from any thin client

The primary building blocks of the proposed design are products and technologies which for the most part already exist within the public domain. Figure 4 depicts the overall system design which is then described in detail in Sections 4.1, 4.2 and 4.3 in the context of the defined requirements. Key design considerations are; the need for a centralized authentication service for all thin clients; profile matching services which provide user sessions with the information required to connect and authenticate to their services; ensuring all services are independent of the virtual machine server by providing a central storage system for all services.

4.1. Multiple services delivered to a single user

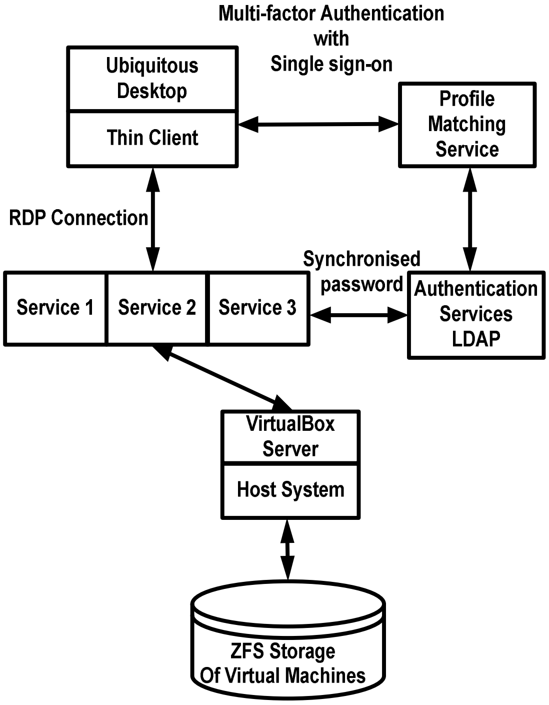

Figure 4. Ubiquitous desktop infrastructure

The profile matching service in Figure 5 provides RDP connection information to the thin client session once the user has been successfully authenticated. The profile matching service will provide the information required to launch a series of RDESKTOP (an open source client for Window Terminal Services) [13] clients within the MilaX thin client. This means that at each login the user should automatically have the RDP connections to their running services opened. Since RDP will not close the service once the connection is broken it is possible for the thin client to disconnect with no state required to be recorded on the client.

As stated already, each service runs on its own virtual machine. There are a number of virtualization server technologies available both commercial and free. The product chosen as part of this design is the Open Source VirtualBox [14] virtualization technology from Sun Microsystems for x86 hardware. VirtualBox allows virtual machines to be created and run on servers in a headless state where the Virtual Machine is accessible from a remote system using RDP. The VM Guest RDP connection is unique to VirtualBox and is a key requirement in this design. Windows 2003, MAC OSX, Linux and Unix virtual hosts are all supported. VirtualBox supports a wide range of guest operating systems [15] (e.g.: Windows Vista, OpenSolaris, and FreeBSD) making it a good choice for this architecture. The Macintosh does not commercially support being installed into a virtualization environment, so the VirtualBox does not officially support this as a Guest OS. The VirtualBox Host Server systems runs exclusively on x86 hardware, and as such all of the operating systems must be run on compatible hardware.

4.2. Multi-factor authentication and single sign-on

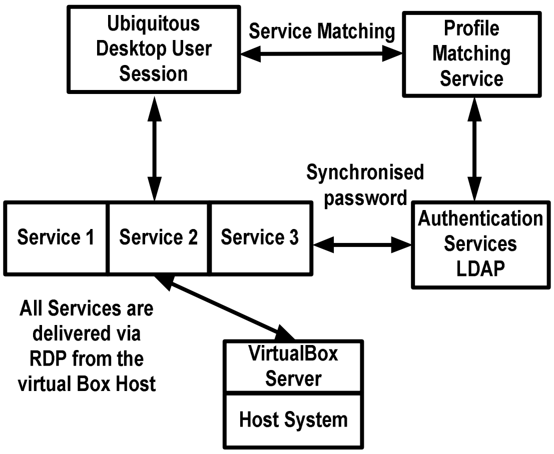
Ubiquitous Desktop User Session
Service Matching
Profile Matching Service
Synchronised password
Service 1
Service 2
Service 3
Authentication Services LDAP
All Services are delivered via RDP from the virtual Box Host
VirtualBox Server
Host System

Figure 5. Service matching ensures that services associated with the user are identified and the authentication information for those services is obtained and presented to the user session.

The authentication and sign-on process is shown in Figure 6. The thin client solution boots and requests a username and password (step 1). The login details and token ID from the USB/CDROM device (step 2) are sent to a UNIX LDAP authentication server running on the profile server (step 3) which compiles a list of available services configured for the user (step 4). The profile matching server sends the thin client all required session details for the RDP connections to their services, including required usernames and passwords (step 5). The thin client establishes RDP connections to the VirtualBox Server (step 6) using the RDESKTOP command where the usernames and passwords are seamlessly sent to the service. The VirtualBox RDP client/server connection uses secure data stream encryption based on the RC4 symmetric cipher.

All authentication methods used for each OS type must be remain synchronized with the profile server which includes details for mounting the central user storage on the ZFS storage device. For a scalable solution, centralized password management is an essential requirement for tracking multiple users and passwords across various services.

4.3. Mobility of the desktop

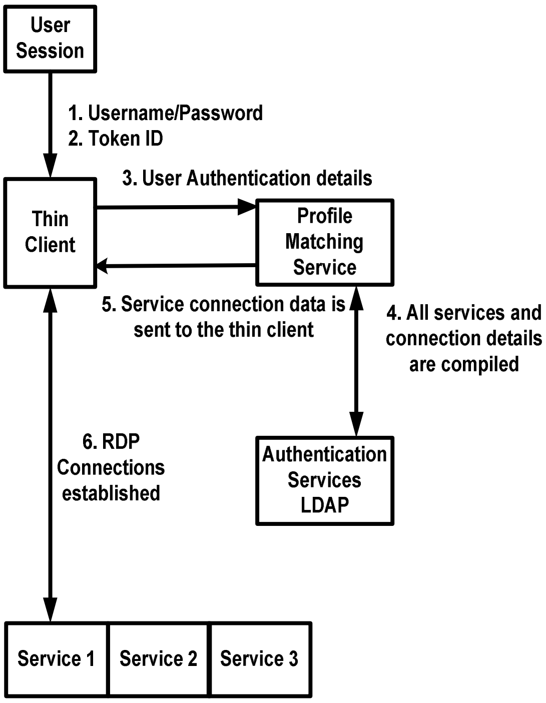
User Session
1. Username/Password
2. Token ID
Thin Client
3. User Authentication details
Profile Matching Service
5. Service connection data is sent to the thin client
4. All services and connection details are compiled
6. RDP Connections established
Authentication Services LDAP
Service 1
Service 2
Service 3

Figure 6. Multi-factor authentication and single sign-on for multiple services, shown over 6 steps.

There are a number of thin client solutions available on the market today. The method chosen is a MilaX based thin client which can boot from a USB pen drive or a CDROM which provides a read only stateless thin client. The MilaX operating system [16] uses RDP clients to access virtual machine guest operating systems. To ensure that the thin clients are stateless and to ensure that no specific server is required to run a specific service, the virtual machines are centrally stored on ZFS which is a scalable high performance storage technology from Sun Microsystems [17]. The architecture of the virtual machines and ZFS is shown in Figure 7. Now that a service is a virtual machine it is possible to decouple that service from a specific VirtualBox

Host. This abstraction from the physical hardware allows future scalability of the computing resources by increasing the number of machines available to run services over time as the virtual machine host systems becomes a stateless commodities used only in the temporary delivery of a service.

5. Conclusion

The need to deliver diverse desktop services to users requires the construction of more complex infrastructures which do not push complexity onto the user experience but rather seek to simplify their experience. The design presented for a ubiquitous desktop resolves key issues inherent in managing multiple services per user. Specifically they include

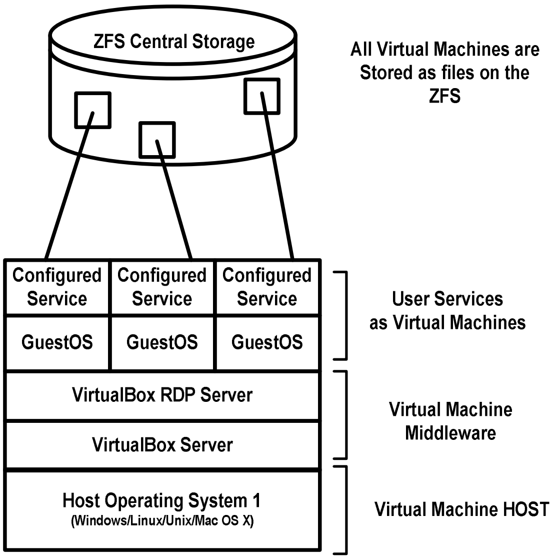

Figure 7. Single sign-on for multiple services

a. Providing multiple services to a user via a matching profile service which centrally manages the relationship between users and their services.
b. Higher level authentication to support the access to multiple services using single sign-on policies.
c. Simultaneous mobility of multiple services between thin client devices.
d. Virtualization of services and centralized storage of services to ensure that the virtual machine host technology becomes a scalable commodity allowing future expansion of the infrastructure.

The authors have implemented portions of this solution within the DIT School of Computing. Virtualized desktops running on a central x4150 server with a ZFS storage pool have been provided for teaching a variety of subjects. A single sign on solution on a SunRay Thin Client accessing either OpenSolaris, Linux or Windows has also been implemented using a consolidated LDAP configuration. A prototype MilaX thin client has also been constructed allowing access to multiple operating systems running multiple virtual machines.

Further research and development is planned on both the infrastructure supporting the ubiquitous desktop and the ubiquitous desktop itself. Specifically; Facilitating high volume service provisioning of virtual machines; High performance matching profile services for large volumes of users; Alternative multi-factor authentication methods; Providing limited offline services on the ubiquitous desktop; The review of privacy within a ubiquitous desktop infrastructure.

6. References

[1] M. Weiser, “The computer for the 21st century,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, 1999, pp. 3-11.

[2] “VMware: Virtualization via Hypervisor, Virtual Machine & Server Consolidation - VMware”; http://www.vmware.com/.

[3] J. Nieh and C. Vaill, “Experiences teaching operating systems using virtual platforms and linux,” *SIGCSE Bull.*, vol. 37, 2005, pp. 520-524.

[4] C. Border, “The development and deployment of a multi-user, remote access virtualization system for networking, security, and system administration classes,” *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, Covington, Kentucky, USA: ACM, 2007, pp. 576-580.

[5] Dong Hu and Yu Yan Wang, “Teaching Computer Security using Xen in a Virtual Environment,” *Information Security and Assurance, 2008. ISA 2008. International Conference on*, 2008, pp. 389-392.

[6] J.C. Adams and W.D. Laverell, “Configuring a multi-course lab for system-level projects,” *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, St. Louis, Missouri, USA: ACM, 2005, pp. 525-529.

[7] B. Villanueva and B. Cook, “Providing students 24/7 virtual access and hands-on training using vmware GSX server,” *Proceedings of the 33rd annual ACM SIGUCCS conference on User services*, Monterey, CA, USA: ACM, 2005, pp. 421-425.

[8] K. Miller and M. Pegah, “Virtualization: virtually at the desktop,” *Proceedings of the 35th annual ACM SIGUCCS conference on User services*, Orlando, Florida, USA: ACM, 2007, pp. 255-260.

[9] “Sun Ray White Papers”; http://www.sun.com/sunray/whitepapers.xml.

[10] “IBM Press room - 2007-11-15 IBM Introduces Ready-to-Use Cloud Computing - United States”; http://www-03.ibm.com/press/us/en/pressrelease/22613.wss.

[11] “Amazon Web Services @ Amazon.com”; http://www.amazon.com/gp/browse.html?node=201590011.

[12] F. Council, “Authentication in an Internet Banking Environment,” *Retrieved June*, vol. 28, 2005, p. 2006.

[13] “rdesktop: A Remote Desktop Protocol client”; http://www.rdesktop.org/.

[14] “VirtualBox - VirtualBox”; http://www.virtualbox.org/wiki/VirtualBox.

[15] “Guest_OSes - VirtualBox”; http://www.virtualbox.org/wiki/Guest_OSes.

[16] “MilaX » MilaX”; http://www.MilaX.org/.

[17] “ZFS: the last word in file systems.”; http://www.sun.com/2004-0914/feature/.