Sorting Through the Hype of Ubiquitous Secure Remote Access and SSL VPNs

[1]Main Entry: **ubiq·ui·tous**
Pronunciation: yü-'bi-kw&-t&s
Function: *adjective*
: existing or being everywhere at the same time **:** constantly encountered **:** WIDESPREAD
- **ubiq·ui·tous·ly** *adverb*
- **ubiq·ui·tous·ness** *noun*

Introduction

If I told you that I could give you a car, exactly like the car you already own with the exception that it gets 1,000 miles/gallon and would have no maintenance costs for 10 years, would you start asking me about adding cup-holders? I doubt it—I sure wouldn't. However, if I told you that I could replace your existing remote access solution with one that provides the exact same capability, but has increased deployment flexibility, increased security options, a lower TCO and faster ROI—I am constantly questioned about whether it can provide "ubiquitous" access. The short answer is "No." The real answer is "No. You shouldn't expect it to, don't really want it to and don't really need it."

Since the days of the first off-box SSL termination solution, I have been contemplating, what at the time I referred to as, the "poor man's VPN". Of course, if I had put any real imagination into it, I would be writing this from my retirement villa in the south of France instead of my cubicle, but that's neither here nor there. The idea back then was that I could access any resource that could be tunneled through an SSL connection from my corporate owned resource without fear of it being intercepted as it traversed the big bad Internet. We played with things like FTP-s and Telnet-s, etc., and as long as we had a client that supported the SSL negotiation, we could use any generic SSL termination box to decrypt the traffic and send it on its merry way to the appropriate backend resource. It worked like a dream and dramatically increased our ability to remotely administer networks without being compromised by the in-security of the channel. While this wasn't an "officially" sanctioned remote access solution and rarely had any strong authentication requirements apart from those of the actual backend resource, it was reasonably secure, protected us from eavesdroppers and still required a fair amount of knowledge—as well as specialized software—to make it work.

Today's world is a whole lot different. SSL-based VPNs are now all the rage from the Fortune 1000 down to the mom-and-pop business—redefining the ease and ability to remotely access resources that were previously only available via IPSec technologies, which often carried their own overhead and problems. This new breed of SSL VPN is far from a "poor man's" solution, many of them providing full layer 3 connectivity to the remote network. Gone are the days of specialized software and potentially complicated client configurations; all one needs is a standard web browser and support for Java or ActiveX. This full featured capability, coupled with the extreme ease of use, has opened the floodgates of ideas about how to exploit this to its full potential—and this usually leads to the concept of "ubiquitous" access. Unfortunately, it isn't that easy, and this concept of 'ubiquitous" secure remote access—remote access from anywhere on any device--is, in my opinion, mostly a pipe-dream useful for theoretical thought, not real-world requirements. I can think of at least three reasons why ubiquitous secure remote access will never, or more probably shouldn't ever, happen; technology, security and business requirements.

The Technology

The first issue with ubiquitous access is the wide range of technology that we must deal with in the real world. There are essentially two methods available today to provide remote access capability. The first, more traditional one used by IPSec as well as many SSL VPNs, is to create a secure "tunnel" through which to send traffic between the remote client and the host network

[1] http://www.webster.com/cgi-bin/dictionary?book=Dictionary&va=ubiquitous&x=0&y=0; Merriam-Webster Online.

device. The second, unique to the SSL space, is to use a proxy to mediate the traffic between the remote client and the host network device. These methods provide varying degrees of functionality and varying degrees of security depending on how they are employed.

Most vendors, including the traditional IPSec vendors who have entered the SSL VPN space, are referring to some type of “proxy” access when they talk about SSL VPN. In essence, the SSL VPN device acts as an SSL terminating reverse proxy, allowing you to access already web-enabled (http/html) content directly from the Internet over an SSL encrypted session. Not to belittle the functionality of these offerings by simply calling them “reverse-proxies”, many can provide advanced features like ActiveX and Java Byte-code rewriting and resigning on the fly as well as some protocol specific proxies, allowing them to provide a greater range of access than a simple reverse proxy. The downside of this method, however, is that apart from a centralized authentication system to provide access to multiple resources and more “depth” in your security posture, these systems don’t provide much more than you would garner from making the actual resource Internet accessible behind a firewall and using the built in SSL capabilities of the native web-server hosting the application. Even more constraining is the fact that advanced web content is increasing difficult to mediate with a proxy engine and the fact that, no matter what you do, a proxy cannot change the fact that the accessing browser may not support that advanced content; “any” browser may not be sufficient.

The more interesting of the two methods is the use of either port-forwarding or a full layer-3 tunnel approach which provides IPSec-like connectivity without the pre-installed client. While these were once called “clientless” VPNs, the moniker has mostly been dropped because all of them require some sort of client-side code to operate, the difference being that most (but not all) provide this client dynamically at the time of access instead of having to be pre-loaded and configured—and the client-side configuration is minimal at worst. This method of remote access provides for any type of resource access (ftp, telnet, terminal-services, file and print, etc.) all without the need for specialize fat clients (like the good old days) and can be a drop-in replacement for most IPSec solutions. In addition, because the client is dynamically installed and updated, deployment time and costs are generally much less than its IPSec cousin. The downside to this method is that in most cases, because of the need of driver level integration to create the tunnel, this method has restrictions on both the OS that can be used and the amount of system-rights the user needs to install the software.

It should be apparent from even this basic information, that the idea of “ubiquitous” access is greatly over exaggerated. In the case of the proxy-based access, you can access resources from virtually anywhere, but you can only access web-enabled content or content that a specific protocol proxy has been written for and every change in even those applications will require adequate testing before deployment. With the tunnel access, you can access virtually any resource, but you can only access it from a supported OS where you have appropriate rights to install the software dynamically—although many vendors support a pre-install capability. Combined together, these provide a wide range of access choices, but choices have to be made and it is difficult to educate end-users to correctly identify the appropriate method on their own. All of this goes to show that truly ubiquitous access should not be expected and can not be delivered in the current state of technology.

Security

So what about your CEO who wants email access at the airport Internet kiosk? As intimated in the previous section, the CEO could probably already have that access—most enterprise mail systems provide some form of web-enabled access and support SSL. Most likely this hasn’t been deployed because of the heightened security risk associated with this scenario as the machine being used to access the resources cannot be verified in advance. The notion of ubiquitous access implies an increase in the number of un-verified access points and presents a serious security issue. The primary concerns are persistent information, and an increased risk of attack either via malicious code or malicious users.

Persistent information, the little artifacts left behind after using a computer to access information, is one of the bigger concerns. The standard browser cache, which can also cache user credentials, as well as temporary directories, are the most obvious concerns with the proxy-based access used in many SSL VPNs. That doesn't mean that tunneled access doesn't also have issues with file-downloads and email attachments that may be cached or saved to the local machine during access, they do. In addition, the now well known "Google Desktop Issue" can also present the opportunity for information to be duplicated and leaked to the rest of the world. All of these persistent items present a serious security threat particularly when the access is from an un-trusted or public terminal.

While persistent information is not a problem solely for SSL VPNs (IPSec can be susceptible to the same issues), the lack of a requirement for a pre-installed client makes it more of an issue. The providers of SSL VPN solutions have given us an arsenal of weapons to help ensure that these artifacts are not left behind. Most of them provide some form of "cache cleaner" which ensures that browser cache, history and cookies are removed from the system when the session is terminated normally and provide some mechanism for cleaning up the system on reboot if the session is terminated abnormally. In addition, many of them provide advanced features that enable you to empty the windows trash can, wipe the temp directory and even to remove its own software from the system. Unfortunately, because each client system and client OS handles these things differently, no cache cleaner is able to work on all client systems and ensure that all traces of persistent information is removed. Some vendors also provide a "protected workspace" or "virtual desktop" functionality which allows the entire session to exist only within this "virtual" arena; once the session is closed, all of that sessions data is eradicated. Here too there are OS limitations and while some of the vendors tout this functionality, they actually rely on 3rd party software that must be pre-installed—not very handy for that Internet kiosk at the airport.

Heightened risk of network attack and infiltration is another issue with wide-spread, ubiquitous remote network access. An unknown system may have viruses, Trojan software, key-logger programs or any other variety of malicious code running on it. You need to know that a) the user's actions and data are not being intercepted and b) the system does not contain any code that could damage your network while it is attached. Since more than one out of every two computer-based attacks is initiated by an internal, authorized person, you also need to be wary of what even your authorized employees may be doing when accessing the system from an unknown location without the watchful eyes of co-workers and management looking over their shoulder.

Again, while these are not new issues only pertinent to SSL VPN, the vendors of these solutions have given us tools to help protect our networks. The "protected workspace" idea previously mentioned can subvert most malicious code by creating a new "instance" that is not running the malicious software; again, assuming that the vendor does not rely on a pre-installed client to make this work. The SSL VPN vendors have also given us a great deal of flexibility in defining characteristics of what we consider to be a "safe" system and only allowing access (globally or by resource) based on whether the client passes these tests. Most of them can test for things like the absence/presence of Anti-Virus (AV) and Personal Firewall (PF) software, the last time these were updated as well as whether it is a trusted vendor for that software. Most can also check things like OS type, OS version and patch level, browser version and patch level, SSL cipher-spec, and a host of other variables. This can go a long way to making sure that the systems used to connect to your network are up to standard. Alas, here too, there are often restrictions concerning the client OS and browsers supported. Many vendors only support one or two AV or PF vendors, as well as often requiring custom code or pre-installed software to get the most protection.

Finally, many of the vendors suggest that their proxy-based access is the ultimate protection against virus and other malicious code from entering your network. While this may be true from a network level, recent studies have shown that most attacks today are carried out at the application layer, not the network layer; almost none of the vendors provide application layer protection or integrated virus scanning of files that are uploaded to the network.

SSL VPN inherently increases your security risks by virtue of its greater flexibility in deployment, ease of use, and breadth of coverage. While the vendors of such solutions—recognizing this fact—have provided sophisticated and unprecedented tools to increase the confidence of its use, the very use of these tools can serve to limit the landscape of acceptable client locations. That is to say, if you take advantage of the advanced security features—as you should—you will most likely exclude many un-trusted clients like airport kiosks. This is the correct thing to do, but in itself proves that truly ubiquitous access is not necessarily a good idea.

Business Requirements

I am constantly amazed at how often I am asked to help solve a business problem and handed a list of technical requirements. Business requirements describe who, what and sometimes where and when; technical requirements describe how. These are two distinctly different things where one is describing the “perfect” solution and the other describes the reality of implementing it. I would posit that “ubiquitous” access, especially that from the proverbial airport Internet kiosk is not a technical requirement, nor— in reality— is it a business one.

I have logged almost 300,000 actual flight miles in the past 3 years. In all that time at airports across the country, I have seen one, maybe two, people ever actually using an Internet kiosk. Furthermore, in all the seminars and presentations I’ve given I have found very few people who admit to ever using one; those that have mostly say that they did it out of curiosity more so than necessity. There are so many alternatives to this today that the usefulness of them— and consequently the requirement to support them— is, in my opinion mostly negligible or at least most certainly a fringe case.

Hanging out with the sales teams and executives who travel non-stop, I know that the real necessity is being able to consistently and reliable allow them to have a VPN connection from hotels, customer networks and wireless hot-spots like those at Starbuck’s and (incidentally) the airport concourse. It is the ability to provide secure access from these (increasingly ubiquitous) public networks that is in question, not secure access from public nodes. The issues encountered with IPSec in even these situations are what has called this technology into question and raised the awareness of SSL-based VPNs.

Not to say that there may not be legitimate business cases for access to secured resources via public terminals or that no one has ever used them for such, what I am saying is that part of the design and analysis process is to refine and define what the business wants and distill what the business really needs. “Ubiquitous access” is much too vague to be a technical requirement and should only be a conversation-started as a business one.

Reality Check

The reasons to investigate, and probably to deploy, an SSL-based VPN are numerous and compelling. They provide easier deployment and ongoing management, more reliable and consistent connections regardless of originating network and a host of new tools to help you provide more robust and appropriate security. All of this drives the numbers of lower TCO and quicker ROI—things the business, and you, should really be concerned with. Of all these reasons though, I hope I have given three pretty specific reasons to prove that a) ubiquitous access is not really possible, b) ubiquitous access is not really desirable and c) ubiquitous access is not really necessary. Holding back on adopting SSL VPN in hopes that it will satisfy this perceived need—or worse, believing the marketing hype and purchasing a solution that promises you ubiquitous access instead of the best set of real-world features you should be concerned with, could prove to be a costly mistake in the long run.

So, how about that car? Are you interested?

About the Author: *Ken Salchow (MCSE, CCNP, N+, C|EH, CCE, CISSP) has been employed by F5 Networks, Inc. for the past five years where he has served in several capacities, currently as a security systems architect. In addition, he is the owner/operator of Binary Forensics, Inc., a boutique computer forensics lab serving the legal community in criminal and civil litigation and Digital Interlopers, Inc., a boutique penetration and testing organization serving small/medium business entities. He currently lives in Minnesota and can be reached at k.salchow@f5.com.*