Security Issues with Distributed Web Applications

Device Connectivity

We are entering the era of “Device Connectivity,” which is the fourth wave of evolution for Internet-enabled applications. The first wave was e-mail connectivity, in which the key applications were e-mail, FTP, and gopher and the goal was file and message sharing. The second wave was Web publication, where the key technology was HTML and the goal was presentation of information. The third wave was created by the adoption of XML and the change in paradigm towards the Web as a programming interface and data-flow enabler. Now, the fourth wave is Device Connectivity, which is achieved by combining various technologies under the .NET framework, such as Windows CE, Simple Object Access Protocol (SOAP), and Universal Description, Discovery, and Integration (UDDI). The goal is seamless access and integration of devices from the production floor, to the boardroom, to suppliers and customers and, ultimately, the individual end-users.

To accurately predict the increase in the number of distributed devices on the network in the near future, let’s review three technology laws that are still valid. These laws are geometric and exponential. They will eventually have to be modified as advances in technology occur; however, they are still valid today.

1) Moore's Law (formulated by Gordon Moore of Intel in the early 70's): States that the density of transistors and integrated circuits is doubling every 18 months. The corollary of Moore’s Law is that computers are rapidly getting faster and cheaper.

2) Gilder's Law (proposed by George Gilder, prolific author and prophet of the new technology age): States that "bandwidth grows at least three times faster than computer power." New developments seem to confirm that bandwidth availability will continue to expand at a rate supporting Gilder's Law.

3) Metcalfe's Law (attributed to Robert Metcalfe, originator of Ethernet and founder of 3COM): States that the value of a network grows with the square of the number of participants. In other words, each additional member of a network adds an incremental amount of value to every other member, thus increasing the aggregate value of the network in a quadratic fashion while the cost-per-user remains the same or even decreases.

While markets are extremely efficient and accurate in the long run, they are completely incapable of keeping up with the pace of technological and business innovations that we currently are experiencing. In turn, as more and more users are connected, the need to increase

information sharing between internal and external audiences continues to increase. As devices become smaller, the user becomes more mobile and the need to change the network paradigm to address transient users becomes more important. Ultimately, all these issues revolve around the need to insure that your model of security addresses these various audiences.

Another major issue concerning this new integrated environment is that the number of connected devices is much larger now. Previously there was only a client, a direct communication link, and the server to consider. Now, there may be multiple clients, multiple servers, and a communication network consisting of many routers, switches, firewalls, and proxies. Each of these nodes contributes their own security risks.

Distributed Network Security

In the past, when access to a remote system used dial-up connection, the need for security measures such as firewalls were not as important. These connections were short and circuit-switched, and the main threat came from war dialers. War dialing (also called scanning or demon dialing) is the practice of dialing all the phone numbers in a range to find those that answer with a modem.

As network technology evolved, devices began to be connected directly and permanently to an Internet. Also, with a higher prevalence of standardized software, an increasing software monoculture, greater connectivity, and more interface points there are more places for things to go wrong. In turn, the significant threats are now from Script Kiddies and Worms. Script Kiddies are defined as the lowest form of crackers, and they run packaged exploit programs against a large number of random targets. Worms are defined as self-propagating security exploits.

With regard to industrial and commercial applications, security concerns are not limited to crackers. These applications must ensure that corporate and manufacturing data remains confidential and safe from unauthorized access and manipulation by both internal and external sources.

The security properties include: Integrity, Authenticity, Confidentiality, and Availability.

- Integrity relates to avoiding unauthorized modification of data.
- Authenticity relates to insuring that the identities of communicating partners are genuine. Security requirements such as data integrity, access control, or masquerade prevention can be served only if one can rely on the authenticity of communicating partners.
- Confidentiality means protecting information from unintended disclosure. Measures such as end-to-end encryption between the connected communicating partners, file protection, and avoiding the publication of proprietary data on the Web are techniques that ensure confidentiality.
- Availability means that systems, data, and other resources are available when needed.

Web Security

In addition to all the requirements of a distributed network, Web applications have further security risks. For example, RPC-like SOAP messaging is less secure because communication does not occur over an existing, secured channel.

Another security issue is the number of untrained users involved with Web content.Although even relatively inexperienced users can set up a Web server and create Web pages, the underlying technologies that enable those tasks are quite complex. These users are likely not to know enough about potential security risks nor have the tools or training to manage those risks.

Despite these concerns, Web applications are essentially client-server TCP/IP applications running over an Internet. We can map Web applications' vulnerability based on security threats mentioned previously.

	Threats	**Consequences**
Integrity	• Modifying user data • Modifying memory or message traffic in transit	• Lost Information • Compromised Machine • Vulnerability to all other threats
Authentication	• Impersonating legitimate users • Forging Data • Spoofing Data	• User Misrepresentation • Belief that false information is valid
Access Control	• Modifying user data • Stealing Information	• Lost information • Lost privacy
Confidentiality	• Eavesdropping on the Net • Stealing info from server • Stealing data from client • Stealing information about network configuration and network traffic • Stealing information about which client talks to server	• Lost information • Lost privacy
Denial of Service	• Killing user threads • Flooding machine with bogus threats	• Disruptive • Annoying • Prevents user from

	• Filling up disk or memory • Isolating machine by DoS (Department of Security) attacks	getting work done

Table 1. Summary of security threat types faced by Web applications (Extracted, with modifications, from reference 1).

To address the preceding issues, the following set of technologies are available:

- Virtual Private Networks (VPNs) address confidentiality and authenticity. Dial-in and Dial-out systems create temporary IP addresses and require a valid return phone number. Dial-in and Dial-out connections are more secure because when the connection is not required or not being used, the device is not on the network. Authenticity can be improved if the server uses a call-back procedure.
- Digital signatures and cryptography satisfy the requirement for authentication and confidentiality.
- Firewalls enforce security at many levels, such as:
- Restricting remote nodes, ports, and protocols that can accept connections
- Accepting packets only from an established connection (packet filtering)
- Dropping packets that do not match established connections
- Dropping packets with impossible flag combinations
- Setting up the firewall to prevent IP spoofing
- Secure Sockets Layer(SSL) insure the secure transmission of information. SSL establishes the authenticity of the client or server, and can encrypt data to insure confidentiality. All Web browsers (such as Netscape Navigator and Internet Explorer) support SSL, and many Web sites use SSL to obtain confidential information.

SCADA and HMI Applications Security

Recently, during an Internet forum discussion, a user stated that security was not a major concern to him because his application is designed to monitor the system only, and remote Web users cannot modify the PLC registers. As he stated, who would invest time and money to break-in and see a bunch of limits on switches?

There is ample motivation for this type of snooping. Competitors (or anyone else) who wants to know the capacity of your company, your recipes, the details of your plant floor layout, the details of making your company’s product, and any other information might attempt such a break-in. Even unauthorized access by suppliers or other users can compromise security. If your Web application is in use, and an important front-door application to your processes, the system

is vulnerable to external attack. These attacks are designed to interrupt service and, ultimately cause a complete shutdown of your manufacturing plant.

Not only is snooping an issue, but spoofing is as well. How do you know the data you receive is not being spoofed? This issue is particularly critical when we are talking about process control and monitoring.

Besides the technologies mentioned in the previous section for securing your environment and your Web-based applications, many SCADA and HMI packages embed security tools that are used during runtime and at the application level. Those tools include:

- IP remote connection filters: Independent of the firewall protection, the runtime can verify the IP address of a remote client requesting data, and enable or disable access based on that IP.
- Dial-in and Dial-out: The dial-in and dial-out can be managed at the application level. The communication channels (including phone numbers, authorized users, and types of information exchanged on those channels) can be defined at the project level.
- View only: The runtime can be configured so that a station does not accept any remote variable modification or configuration allowing it to provide only replies to data requests.
- View only, according to user or status: The same protection described in the preceding bullet can be configured in the application layer to disable remote modification of variables and commands, or to provide information based on the user's priority.
- User authentication: Username and password can be applied at the application, screen, or data level.
- Object protection: Some packages allow object-property definitions based on process status, remote user IP, and their IP address. These properties can restrict whether the object can be viewed and edited.

Conclusions

More devices, more users, and new architectures all equaling greater access and creating more opportunities for a breach in your network security. As we discussed here, there are technologies available to address security concerns, but first you have to be aware that there are concerns and second, which security issues apply to your network environment. Do not assume that your applications and your network will not be affected by or are of no importance to snoops and crackers... it could be a costly assumption.