GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials Bootcamp Style (Security 401)"
at http://www.giac.org/registration/gsec

Using Windows XP to Secure a Learning Tool Dan Rushton, 7/14/03
GSEC Practical Assignment, v 1.4b, option 2

Contents

1) abstract
2) security
3) planning
4) implementation
5) conclusion
footnotes, references and appendices

© SANS Institute 2003, Author retains full rights.

I Abstract

The goal I have been given is to secure a connection between a Mitsubishi RM-101 programmable robot training arm ("the arm") and the internet so as to be publicly available as a learning tool. In addition, it must be scalable and portable so that when funding is added (or cut), it can be expanded (or moved to the basement) and still be available to those who wish to learn. For the purpose of this paper, and to satisfy the "scalable" specification given above, I shall treat the existing connection as a generic ISP and secure my small network as a stand-alone. If necessary in an extreme case, I can use one of the old modems I rescued from the trash and firewall the dialup connection. Actually plugging it in and convincing it to work took a few days. Planning and securing it, and writing this paper, took longer.

Fortunately for me, WinXP greatly simplifies the setup of a small network. All I have to do is make a crossover cable from a regular LAN cable[1] and open the Networking icon in both ends, and in a few seconds the GUI tells me it is making a connection and finalizing communications. Unortunately for me, it does a LOT of the configuration automatically, which means for one thing that it can (and often DOES) change connection settings to suit it's own needs. In order to keep me from inadvertently changing settings, many of the features and functions must be configured and disabled in several places.

The whole idea is going to be tying together some of the tools available to me to secure a small network, and, in the worst case, track down what happened to it. What follows is my attempt to add a small amount of clarification to the wealth of security knowledge available.

II Security

The security considerations for an EDUcational institution are somewhat different than for a COMmercial institution, in that much more of the information safety is in the hands of the user. I personally feel that in a learning environment, information freedom is more important than information control, so the preservation of our "crown jewels" of data and hardware will rely much more heavily on redundancy and backups than on "locking it up" to keep it away from everyone. More specifically, the information security policy already in place is, in short, to "...protect its information resources while supporting the relatively open access required by academic pursuit... " as set forth in this institution's IT security home page[2].

In keeping with the "Defense in depth" strategy[3], I have the power to restrict information coming in to my network to a specific set of strictly ASCII characters, and therefore have the responsibility ("With great power comes greater responsibility") to efficiently parse the strings and check for out of range values so the arm does not damage itself. As a future feature, someone in a programming class could work on an interface.

Due to time and length considerations, the only tools and procedures I will focus on are;

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Logging with Windump, XP ICF logging (dual logging to help meet the redundancy specification above), OS slightly more secure settings.

Some of the physical security concerns are; to keep maintenance personnel (like me) from being injured by unexpected movement of the arm when someone sends a remote command to it, and to keep the arm from damaging itself or the tools it uses. To meet these requirements, I can implement things like: command buffering for offline maintenance, command parsing and bounds checking, exception handling for out of range values and anything unexpected or just plain invalid (better covered in a different paper).

III Planning phase:

The most important specification for any hardware and software that is to be included in my system is: FREE AND PUBLICLY AVAILABLE. This includes; freeware (and things that I can write and declare as freeware), anything that someone else has released control of (thrown out and given me permission to salvage) and I can repair, things already owned by the educational institution that I can “borrow”, and things that I own and can afford to relinquish control of or replace. Additionally, the final instructions to the arm must be available in *strict* ASCII format, not printer strings and controls.

The equipment I have to complete my goal that meets these specifications;

1) the arm
2) a digital camera
3) three identical computers, slightly less than minimum system recommended by XP (table 2 for those who are interested)
4) the operating systems that this educational institution already owns
5) some of my programs
6) some freeware
7) anything else I can scavenge

I know the arm already works, and I have the specifications (table 1) for the instructions it needs to carry out commands. Additionally, I would like very much to learn more about the features and functions of Windows XP, so I have decided to use one of the computers I have salvaged as the firewall for my small scale network.

In designing my network, I began with three machines, all identical hardware, and some homemade crossover cables (no hubs). I wish to use one as a firewall, one as a server, and one as the driver for the arm. I have considered using one machine for all purposes (and probably will as my machines get reposessed), but this keeps things compartmentalized in my mind and gives me more networking experience.

Why all the bother of creating a new network, when a perfectly good one is already in place? Because it simplifies several aspects of implementation. For example, because it is an educational network, I do not have the liberty of simply restricting access to information or systems I feel like (the "open access" part of the security policy) as freely as I would be able to if I owned a company and could claim intellectual property rights. Furthermore, I get the luxury of

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

reconfiguring and/or reinstalling on a whim when I am the sys admin **and** the boss.

According to Microsoft[4], a firewall is recommended when a single computer is "connected" to the internet (Note: by "connection" MS means the software connection represented by the icon in the "Network Connection" section of the control panel, not a physical connection. Maybe I've been playing with hardware too long, but this is still a source of confusion to me). A firewall should not be used for VPN connections or any connection that does not directly connect to the internet.

This, of course, assumes that I have a LAN card (or other hardware) dedicated to the internet connection, and all other computers are connected by some other hardware, often a second LAN card handling the router or hub. In my case, an ethernet adapter for the internet connection to the firewall, a parallel port cable connecting firewall to server, ethernet cable connecting server to the driver for the arm, and parallel port cable connecting the driver to the arm.

When the default firewall log exceeds its alloted size, the information in that log file is written to a new file with the same name and a ".1" extension appended to it. I would like to squirrel this information away as quickly as possible, so I will either poll the existence of this file name (which I hope doesn't take up a lot of resources), or preferably, use its creation as an event trigger to move that log and rename it. I will also find out if the firewall supports renaming the default logs, and how receptive it is to it.

So, naturally, the first thing I want to do is enable the firewall on my existing connection to the other computer and "bang on my own door" to see what happens in the logs. Unfortunately, the logs reported an awful lot of information before I even connected to the other computer, some of which I was able to discover and fix in the implementation phase.

Folder security options for ALL USERS:

Automatically search – off, show hidden – off, hide extensions – off, hide protected – on, show control panel – off, show encrypted – off, use simple file – off (until I find out why I should turn it on)

Caveat: simple file sharing initiated without the wizard turns the Guest account back on[5], uses it for all network logons, limits my ability to configure sharing options and permissions. There is in fact a way to activate the guest account from the command line with the **net user** command:

```
net user guest/activate:yes
```

This form does not add the Guest account to the welcome screen, but does remove Guest from the list of accounts that are denied access to the computer from the network, so I'll be checking that account fairly often.

Information and resource hiding: increasing the cost of data mining. A tactic I have used in the past is social "reverse" engineering script kiddies: find out what bores them and put my sensitive stuff there. Others include 1) randomization of hiding schemes and methods, updating these methods often (definitely the most time and labor intensive, but can be worth the effort). 2) create some interesting places to hide things: fractint[6] and The Sims[7] have been some of my favorites in

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

the past. Fractint uses incremental file names by default, and they are all the same size and extension (gif, bmp or png) and when I compressed my logs and changed the extensions, they were nearly indistinguishable from the real thing. The Sims has much larger files, and miscellaneous file sizes so that variations in file size were more difficult to detect, but the skin files became quite popular for a while, so I had to leave those intact to make it look realistic. Some of the nice things about using a game to hide files were that when someone tried to run it, the game asked for the original CD (as long as I had left the executable intact), and if someone actually did decide to copy one of the files that was actually my log and tried to use it, it didn't work and the game was not helpful enough to give them the clue that the extension was changed. At which point almost anyone would delete it assuming it had become corrupt in transit rather than expend any effort trying to fix it.

I hope that making my system more stable against probing does not encourage brute force attacks, but it will, so I might as well log them with things like the firewall logs and Windump.

IV Implementation

BUT, when I actually plugged the arm in to the parallel port of a Windows machine and gave it a command, the arm promptly choked. This made me curious as to what is being sent to the port, so I used a parralel port monitoring tool[8] I have used before and found something very disappointing: sending characters to a parallel port through windows carries a LOT of printer setup commands, none of which the arm can successfully process. It seems that Windows of any flavor REALLY wants to treat anything connected to the parallel port as a printer complete with control strings, until I found the proper combination of printer setup options in XP: in the printer control panel I have set it to "generic" printer, and "text only" output.

The installation of XP is heavily dependant on the incumbent OS. For example, I have found out since actually installing and booting it, that my motherboard/BIOS won't boot from the CD and XP won't install in DOS, so I've had to install W9x as interim to XP. The W9x install directly affects the registry process of XP, and it can be very time consuming to "install enough" of 9x to be able to use the ethernet adapter efficiently (despite the fact that XP maintains that it doesn't support an upgrade from W95, that it has to overwrite the system files).

During the many installations I forced myself to do, Windows touts itself as "the easiest Windows yet", and I believe them. Some of these "easy" items that worry me are; remote assistance, the system preparation tool, unattended setup runable from remote machines (I know, a few thousand network admins just shook their heads in disbelief of my nievetae, but it still worries at me), file and settings transfer wizard (and at least five other wizards), user state migration tool, .NET passport integration (seems like a good opportunity for password broadcasting from its description in Help and Support), remote desktop, and simple file sharing to name a few.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Following the guidelines of “Keep it simple” and “Security in depth”, I begin by choosing to NOT install anything the connection absolutely will not need. This results in the OS kernel and accumulated patching using “only” 1.2Gb of space on my hard drive to install itself (call me paranoid [and I am], but this seems like either a bad case of overkill, or a large margin for error). Due to time restraints, I leave it as another avenue of research for further papers as to how many of the 10,400 initial files I can dump and still have a viable OS (better covered under a future paper on SuperTweaking XP), although that many files is attractive as far as naming an important file (eg. logs) to one of those to hide my needles in a haystack.

Default install, absolute minimum by selecting only TCP/IP options. Then, start turning things off[9];

1) MSN Messenger – “open” from systray;
“cancel” .NET password signup (if it surfaces before and/or after service pack and other patches), in “tools… options… preferences” uncheck “Run … starts” and “Allow … background”. I personally will, in “Personal”, keep “display alerts …” on to be notified when a software package decides to reinstall it someday, and “Always ask me for my password …”.
2) Guest account is turned off by default (see planning caveat, above).
3) Created as useless an “Administrator” account as I could, and hid the real admin as well as I could.
4) To make it more difficult for an installation process to convince the machine to change settings, under “connection settings”, a) network connection was set to “Authenticate as computer when computer information is available” so I turned that off, b) was not set to “Authenticate as guest when computer information is unavailable” fortunately, c) was set to “Enable network access control using IEEE 802.1x” with EAP types available: “MD5 challenge” or “Smart card or other certificate” (under “properties” for Smart Card are 103 possible “Trusted Root Certificates” to choose from, but under MD5 challenge “properties” there are 0).
5) Taskbar and Start Menu properties are set to “Hide inactive icons” in notification area, which can make it useful for me to audit system use.
6) In “System Properties”, “System Restore” and “Remote Assistance” are on by default.
7) Dump the restore point created by the service pack and patches, if applicable.

I have discovered that XP is very good at running many DOS commands, even if it does them it’s own way (previous students claim that many of the low level interrupts have been remapped in XP, yet another branch of further research). Even better, they have kept many of the NT/2000/2003 commands as far as I can tell, and have apparently added new ones and expanded a few old ones for XP[5](someday I’ll write another paper on this line of thought as well, if no one beats me to it).

The communications overhead is horendous, and I have noticed some inconcistencies in logging: for example,with QOS Packet Scheduler installed and while running Windump I have set it to close after catching 20 packets, but it is

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

still capturing after the Local Area Connection reports having sent 84 packets. Neither Windump nor the connection on either end report any dropped or missing packets. In fact, my setup is reporting the server broadcasting 84 packets just to initiate a connection, but the connection on the other end does not report receiving any. Further, when the OS is installed without any connection protocols, it sends 1 packet at startup, and when the TCP/IP communication protocol is added, it starts sending packets every few seconds with no one receiving any. This leads me to believe that either: the OS is sending packets over several types of connections, or the kernel is giving much higher precedence to its connection processes than the Windump process, resulting in Windump "missing" many packets.

All this observation and speculation resulted in my taking these steps to find out what IS going on in there;

1) try to capture all packets from that connection
2) learn more about the loopback adapter displayed by the route command
3) learn about how XP uses ARP broadcasts

Using windump with the –D option (specific to the Windows version) returns a text string of devices it finds available at the moment. This I send to a text file for later reference. Advantages: 1) I have a list of devices windump has found 2) if the file is empty, I won't bother running windump because it has nothing to sniff from. I can just have it keep polling until it finds something. Disadvantage: everyone else has a list of devices windump has found until I hide or delete this file after I use it.

By using either a script or a small executable included in the list of things to run immediately on startup, I can poll all my interfaces until something shows up, save it's identifier in a file, and use that string to call windump to start sniffing (future research: are the number of instances of windump I can run concurrently limited more by my small amount of RAM, or does the multitasking capability efficiently increase it to the stack limit of XP?)

Using the **route** command, I can view and set the network routing tables. XP definitely sets these dynamically, apparently rebuilds as much of them as I let it every time it boots, even if I set them as static (maybe I'm just doing it wrong). XP also gives an indication of those hard-to-track down problems with hardware when it *insists* on rebuilding or changing the IP connection stats on every boot, or, especially, during operation. But, if using the "repair" choice in Network Connections gives the error "failed to renew IP address", I must suspect the inability of the software connection to communicate with an external router (like when I unplug the Cat5 cable).

Another caveat: due to the extremely helpful nature of XP, if a program (such as Internet Explorer) has a problem using a connection that is and should be firewalled, it may well create a new connection to the same hardware, which bypasses the firewall until one is enabled on the new connection that the application is using. From my experience on a broadband help desk, multiple spontaneous connections to the same hardware indicates a hardware problem, but if a connection is bypassed only once it indicates a configuration problem

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

between the connection hardware, the firewall, and/or the application using the connection.

Route in XP reports two interfaces, windump (using the –D option) reports only one – the hardware. I will continue research into whether some of these packets sent are actually to the loopback interface and whether windump sees these also, or whether the continual broadcasting is something "uncontrollable" like the Hello packets for router table configuration, but Windump should be seeing those.

In trying to utilize the stateful inspection firewall included with XP Pro, I followed the suggestions of Matt Snitchler[10], et al. to manually configure it so I can concentrate on the logs.

The "Help and Support Center" is far too helpful for my tastes. I have found a few settings that change for me by simply starting the help center and reading a few help pages. For example, the network settings "Authenticate as computer…" and "Enable network access control…" I have specifically turned off, but after clicking through some of the help pages I have found them turned back on, and this phenomenon is repeatable (yes, I do remember the part of the End User License Agreement that translates to something like "…we assume no responsibility for your ignorance in using the Software…" but gimme a break!). Another bug in my shorts, telnet is available by default as soon as IP protocol is enabled, even after the patches.

ICF logging is flexible as far as where I can put the temp files, and it renames the filled log to *.log.old (rather than the previously reported *.log.1), so by polling for the existence of this .old file I can hide it sooner, and by choosing the file size to be smaller (down to the minimum 1kB) I can save it more often (another update to the help files, XP does not save the old logs sequentially as implied by the .1 extension, but immediately overwrites the previous log, so I have to copy or move the completed logs more often than I had planned). Also, I haven't yet found a "verbose" viewing mode for the ICF logs, but I'm still looking.

So, what is in those logs? I finally gave myself some time to check the event log viewer with the resultant "Ick!". Apparently, XP wants a service to connect to Microsoft's servers at startup, which results in a lot of packets reported by the network connection status count and firewall logs that look like this:

```
2003-07-01 09:53:20 DROP UDP 0.0.0.0 255.255.255.255
      68 67 330 - - - - - - -
2003-07-01 09:53:29 DROP UDP x.x.x.x x.x.255.255
      137 137 96 - - - - - - -
```

At first glance it looked like the firewall was dropping its own DHCP requests from the number and frequency of the 0.0.0.0 source addresses, but I later discovered many of them to be automated requests to a Microsoft time synchronization service gone bad, and it wasn't until I checked the event logs and found several failures of W32Time that referrence NtpClient and finally led me to the Windows Time service as the culprit. The source and destination addresses had me confused because the DNS table can't update until I actually hook it up to the internet. This spawned a long list of services to stop and disable under "control panel – Administrative tools – Services". By right-clicking a service and choosing "properties", a drop-down list usually lets me choose

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

between; automatic, manual, or disable. I started by disabling the things I just don't like the sound of (28 of the 77(!?) default services on the first round and counting), reboot and run a few of my key programs to check stability, and continue disabling or at least switching from automatic to manual until I get the system somewhat minimalized. I ended up with almost exactly the configuration recommended by Black Viper and his cats[11], with a few nods to hardware and personal preferences. After rebooting and noting the performance change, checking the services just moved a LOT higher on my list of things to do when I install a system.

Incidentally, when I turn on 2 logging types, the camera refresh rate drops from 3 frames per second to one every 2 seconds, among other performance hits (a direct result of the previously mentioned hardware specifications).

V In conclusion, once the construction clears we can connect the arm to the internet and ask people to write controlling strings to have it do things. At the time I am finishing this paper, I am down to one machine to use as firewall, arm driver and log archive, and the lab I am putting the nework together in is under construction and being moved to the theatrical department.

Unanswered questions

How far did MS go in remapping their own low level interrupts since DOS ver. 6.22?

How many W2000/03 functions are in here?

Is my XP serial number in the time sync packets?

Why so many active UDP connections?

How many of these services can I turn off, and which are dependant on connection type?

Which registry keys do I change so I can get XP to re-initiate the USB camera without having to reinstall the OS?

Can I get XP to stop rebuilding the routing table on boot?

How can I disable some of these ever-helpful wizards?

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Footnotes, references, appendices:

[1] Crossover cable diagram
http://www.netpilot.com/uploads/42/CABLES.pdf

[2] security policy
http://www.its.mnscu.edu/security/policy/index.html

[3] "Security Essentials", Steven Northcutt and Eric Cole, 1999

[4] "MS Help and Support", Windows XP

[5] "Windows XP Tips & Techniques", Walter Glen and Rowena White, 2002

[6] fractint, Bert Tyler et.al.,
http://www.fractint.org

[7] "The Sims", Electronic Arts Entertainment, 1999
http://thesims.ea.com/

[8] parralel port monitor, Mark Russinovich, 1998
http://www.sysinternals.com/ntw2k/freeware/portmon.shtml

[9] "Protecting XP from intruders", Chris Neglie, 2003
http://www.tweakxp.com/tweakxp/display.asp?id=1736

[10] SANS Reading Room:
"Introduction to the Microsoft XP Firewall", Matt Snitchler, 2001
http://www.sans.org/rr/papers/67/278.pdf

[11] "Windows XP home and pro service configuration", Black Viper,
http://blackviper.com/WinXP/servicecfg.htm

[12] windump, Piero Viano, Loris Degioanni et.al., 1999
http://windump.polito.it/install/default.htm

"RM-101 user's manual", Mitsubishi Corp, 1984

"Networking with Microsoft TCP/IP", Drew Heywood, 1998

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Appendix

Table 1	control strings for Mitsubishi RM-101 (short form)
Command string	expected result
F	Open gripper to Fullest extent of limit switch
C	Close gripper to limit switch
H	Set current position as Home
N	Nest = move to last set home position
I 1,1,1,1,1,1	Move motors in Incremental steps from current position
P1 1,1,1,1,1,1	Define a Position relative to current home position
M1	Move to a defined position

Table 2 computer hareware

Dell Optiplex Gxi
Intel Pentium MMX, 166MHz, 64MB RAM
3Com 905TX ethernet adapter
2GB hard drive
CD ROM and floppy drive (reposessed after last installation and patching)

© SANS Institute 2003, Author retains full rights.

Last Updated: April 26th, 2017

Upcoming Training

Click Here to
{Get CERTIFIED!}

SANS Riyadh 2017	Riyadh, Saudi Arabia	May 06, 2017 - May 11, 2017	Live Event
SANS Security West 2017	San Diego, CA	May 09, 2017 - May 18, 2017	Live Event
SANS Northern Virginia - Reston 2017	Reston, VA	May 21, 2017 - May 26, 2017	Live Event
SANS Melbourne 2017	Melbourne, Australia	May 22, 2017 - May 27, 2017	Live Event
SANS London May 2017	London, United Kingdom	May 22, 2017 - May 27, 2017	Live Event
SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
Community SANS Philadelphia SEC401	Philadelphia, PA	Jun 19, 2017 - Jun 24, 2017	Community SANS
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS