Global Information Assurance Certification Paper

The Real Effects of the Microsoft Hack-in
Stephen J. Lopez
November 13, 2000

Introduction

Most security architectures, procedures, and policies are centered on the prevention of real-time hack attempts. Traffic monitoring of external network connections, suspicious traffic identification, and unusual login attempts are all normal approaches to intrusion detection employed by most network security architectures. The typical hacking scenario involves an intruder on the outside of the organizational security perimeter defenses trying to probe or exploit operating system weakness or hardware vulnerabilities. Firewall products and security appliances are becoming much smarter, making a successful penetration of security defenses more difficult. These advances in perimeter security applications are forcing hackers to find more creative exploit and penetration techniques. With the discovery of the hack-in of the Microsoft corporate network on October 17th, 2000, the hacking scenario was forever widened exposing a clever hacking technique that most likely could have been prevented.

At the center of the Microsoft hack-in is a clever attach methodology that completely bypassed the perimeter defenses of the Microsoft network. The hackers attacked the computer of an employee with remote access privileges by using an easily detectable trojan "backdoor" program instead of directly attacking the perimeter defenses of the corporate network. The trojan backdoor, TROJ_QAZ is quite dangerous in that it allows hackers to access and control an infected system. The TROJ_QAZ trojan was initially distributed as "Notepad.exe," but can appear as different filenames on an infected system. The malicious code attempts to spread itself to other shared drives on local networks, which is what makes this hack-in so unique. The hacker successfully choose an entry point (asynchronous dial-up) into the Microsoft network that was most likely unguarded and vulnerable.

A Check-Point Failure

When an infected file with the TROJ_QAZ trojan is executed, the worm registers itself in the Windows registry in the auto-start section of the infected system. The worm then stays in the system memory of the infected computer as an application (visible in task list) and runs two processes: spreading and backdoor. The ability of the trojan to establish itself on a workstation that is connected to the corporate network is the first failure in the Microsoft security architecture.

The TROJ-QAZ trojan behaves like a virus with a detectable virus signature. Anti-virus software should have been active when the employee connected to the corporate network

and should have been able to detect the worm and eradicate it. On an infected system, the original NOTEPAD.EXE can be found with the NOTE.COM name (it is used by the worm to run the original Notepad when the worm completes its routines), and the worm's code is present in the NOTEPAD.EXE file. The virus signature for the trojan was identified in July 2000 by Symantec, ironically one of two anti-virus vendors contracted by Microsoft to protect the Microsoft network from virus infection.

As a result, the virus signature should have been present in the anti-virus program signature DAT file on the remotely connected system (assuming the employee updated the DAT file regularly). If the Microsoft security architecture was more effective, an automated virus signature file update procedure should have been triggered for every system that remotely connects to the network. Such an automated procedure, which is simple to implement, would have most likely caught the trojan code before it had time to spread any further into the corporate network.

The migration process of the TROJ-QAZ trojan spreads the worm through the local network drive connections to data areas accessible on the network using the employee's user ID. These drive connections are shared for reading/writing data to the network and could possibly contain many instances of the replicated worm code. Because of non-existent virus scanning at this network entry point, the intruder was able to gain access and "roam" around undetected in the Microsoft corporate network for three months or more, increasing the ability of the worm to replicate itself across the network.

The Consequences

What makes this incident so important to the security community is that the Microsoft hack-in is the first reported incident where an outside hacker was able to penetrate a corporate network and maintain connectivity with the target for an extended period of time. There is much more at risk than public relations, perception, or image. Microsoft has

alleged that this hack-in could have been industrial espionage. If this is true, this hack-in could have much more far reaching repercussions, such as infecting product source code before release to manufacturing. It is clear from the official press releases from Microsoft that the hackers gained direct access to program source code. Could the contamination of product source code have been the actual purpose the hackers targeted the Microsoft network?

Microsoft officials allege that the source code was not altered during the hack-in, but admitted that hackers did indeed view source code for at least one product. It must be assumed, since the source code was "viewed", that the source code was copied. Microsoft admits that the trojan was sending e-mails to an IP address outside the US. However, since the hackers have seen product code, it can be assumed that the basic logic of the source code is better understood by the hackers. The result of this hack-in will most likely be better hacking tools for Microsoft operating system.

The advantage that Microsoft operating systems held over UNIX operating system products was open source versus closed source. Since Microsoft operating systems were essentially closed source products prior to the hack-in, hackers had to look for vulnerabilities in Microsoft products through trial and error. It is irrelevant if the source code was altered within the Microsoft network or even copied by the hackers. The critical point is that the hackers most likely achieved a better understanding of the source code programming logic by achieving the ability to "view" the source code over an extended period of time.

It is conceivable that the Microsoft corporate network was not the actual target of the hack-in but that Microsoft's customers were. The possibility of infecting a Microsoft product with a backdoor trojan that could be activated after the software is shipped to millions of unsuspecting and unprotected users is very alluring to a hacker.

The backdoor routine of the TROJ-QAZ trojan is quite simple and easy to embed in another program. The trojan code supports very few commands: Run (to run specified file), Upload (to create a file on affected machine) and Quit (terminate the worm routines). With these three commands the ability exists to install the trojan to any other backdoor (possibly another trojan or virus) on the infected system or to the network the infected system is connected. The worm also sends a notification to its "host" (possibly the original author of the hacker that embeds the trojan), through an e-mail message that contains the IP address of the infected machine.

Conclusion

The hacker realized that it is much simpler to target a less secure, virtually wide-open system of an employee of Microsoft, then it would be launching a direct attack on the perimeter defenses. It is this approach that makes this hack-in particularly insidious. Most organizations provide remote access to the corporate network for employees via asynchronous connections. Since most employee systems that connect to Internet Service Providers (ISPs) barely use virus protection let alone contain firewall capabilities, it becomes very easy to place a trojan through an unprotected employee connection onto a targeted network. Since most organizational security architectures are designed for frontal assaults on the security perimeter, the remote access "flank" is most likely left wide open.

At issue with "flank hacks" that attack the flank of organizational defense architectures is file and system access. Since the main incentive for most employees to dial-into a corporate network is to work from home, user names, passwords, data, and programs are at risk. In the Microsoft hack-in, the TROJ_QAZ specifically attacked the shared folders and network resources accessible by the targeted employee. It is unclear at this point in the investigation if the employee was specifically targeted because of perceived access or if the target of the hack-in was the source code repository of Microsoft products.

It is unlikely that the ultimate goal of the hack-in will ever be discovered. However, one often overlooked security issue is perfectly clear. As more of the work force migrates to telecommuting, the potential for more worm based industrial espionage from the flanks of the corporate network in the future is a major risk to the overall security of all corporate networks.

References

McAfee J., Haynes, C.. (1989). Computer Viruses, Worms, Data Diddlers, Killer

Programs, and Other Threats to Your System. New York, NY.: St. Martin's Press.

ZDNews. "New account of Microsoft hack-in". ZDNetNews. October 29, 2000.
URL: http://www.zdnet.com/zdnn/stories/news/0,4586,2646402,00.html?chkpt=zdhpnews01. (14 Nov. 2000).

Berinato, Scott. "Is Microsoft hack a wake-up call?". eWEEK. October 27, 2000 .
URL: http://www.zdnet.com/zdnn/stories/news/0,4586,2646167,00.html. (14 Nov. 2000).

Berinato, Scott. "Security Experts Probe Microsoft Hack". eWEEK. October 27, 2000.
URL: http://www.zdnet.com/zdnn/stories/news/0,4586,2645873,00.html (14 Nov. 2000).

Lemos, Robert. "MS Intruder may Elude Authorities". ZDNetNews. October 27, 2000.
URL: http://www.zdnet.com/zdnn/stories/news/0,4586,2646331,00.html (14 Nov. 2000).

Bowman, Lisa M.. "MS attack takes hacking to new levels"ZDNetNews. October 28, 2000.
URL: http://www.zdnet.com/zdnn/stories/news/0,4586,2646315,00.html. (14 Nov. 2000).

Lemos, Robert. "Microsoft -- burned by anti-virus tools?". ZDNetNews. October 27, 2000.
URL: http://www.zdnet.com/zdnn/stories/news/0,4586,2646200,00.html. (14 Nov. 2000).

Foley, Mary Jo. "MS hack: Was source code altered?". ZDNetNets. October 27, 2000.
URL: http://www.zdnet.com/zdnn/stories/news/0,4586,2645871,00.html?chkpt=zdnnstop. (14 Nov. 2000).

Bridis, Ted., Buckman, Rebecca. "Microsoft hacked! Code stolen?" WSJ Interactive Edition.
October 27, 2000.
URL: http://www.techtv.com/screensavers/showtell/story/0,3656,3008126,00.html. (14 Nov. 2000).

Viruslist.com. "AVP Virus Encyclopedia – Internet Worms: Worm.Qaz". URL:
http://www.viruslist.com/eng/viruslist.asp?id=4077&key=00010001301000300001. (14 Nov. 2000).

Abreu, Elinor. "Dutch hacker breaks into Microsoft Web server again". The Industry Standard. November 16th, 2000.
URL: http://computerworld.com/cwi/story/0%2C1199%2CNAV65-663_STO53648_NLTs%2C00.html. (15 Nov. 2000).

Last Updated: May 30th, 2017

Upcoming Training

SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
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
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
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 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
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	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 Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event