Global Information Assurance Certification Paper

Learning from what Intruders leave behind

John R. Dysart
12/29/2000

Introduction:

The lessons learned from intrusions tend to focus on how the intruder got in (the vulnerabilities and the exploits). When people discuss what an intruder leaves behind, Trojans, Back Doors, and other "hacker tools," recent discussions tend to focus on Windows based computer systems and the growing threat to individual PCs. For example, the SANS GSEC course discusses three Trojans (back orifice, netbus, and sub-seven). Windows based computer systems are the primary target for all three of these Trojans. The recent focus on Windows based Trojans is understandable. Trojans targeting Windows based systems are far more common. Simovits Consulting's web site lists 347 common port numbers for Trojans and ONCTek LLC, lists 219 common port numbers for Trojans/Backdoors. Simovits Consulting also lists these Trojans by operating systems (target environment). This listing shows approximately 35 Trojans for UNIX variants and over 400 for the Windows operating systems. Further broadband access, via DSL or cable modem connections, has become much more common. This means that many people with only minimal computer experience and no experience with computer security now have a permanent "static" presence on the Internet. Fortunately, the recommend protective action, installation of a personal firewall, is relatively painless. The tremendous amount written about programs an intruder might use to identify or exploit a vulnerability is also understandable. Given a choice between preventing an intrusion and minimizing the damage caused by an intrusion, we would all choose to prevent the intrusion. Unfortunately this is not an either or situation, despite the best intentions intrusions do occur. With the exception of password cracking programs, far less has been written about what can be learned from the programs left on UNIX systems and much of what has been written goes much beyond listing the port numbers commonly used by the programs.

The purpose of this paper is to look at and learn from the programs intruders install on UNIX systems after the initial compromise. This will be done in four parts sections. The first will discuss commonalities or trends from a number of recent incidents. The second section will discuss what lessons these tools teach system admistrators and others responsible for the day to day security of computer systems. The third section will look at law enforcement and computer security personnel can learn from these programs. Finally, we'll look at what light these programs might bring to current arguments about proposed laws to address computer crime.

While this is certainly is not a scientific survey, over the past year I have had the opportunity to work on half a dozen cases where programs installed by an intruder were found on UNIX systems at large companies, ISPs, and universities. In reviewing these

cases and the tools found on the victim systems a number of common themes have emerged. Some of these themes have important implications for the choices system administrators make on a daily basis. As such they are valuable "lessons learned," even if the lesson is only that you really need to do some basic activity, like encrypting passwords and having a good password policy. Other themes raise interesting questions about both our current laws regarding "hacking" and some of the legislation and treaties that are currently under consideration.

The victim operating systems included Linux, Solaris, and IRIX (SGI).

In each case several tools/unauthorized programs were eventually found on the victim system.

In 5 of the 6 cases, the victim was unaware of the intrusion for a prolonged period (over 20 days).

The three most common functions of these programs were packet flooding, packet sniffing, and "backdoor" programs.

Most of these programs had to be complied on the victim system, and the most dangerous of them needed to run as root or UID 0.

The victim computer system was either outside the organizations firewall, or in a DMZ. In some cases there was a good operational justification for the lack of a firewall, in other cases there was not a reason.

The programs were designed to be "script kiddie" friendly. They had good documentation and help functions. In the two cases where the intruder has been positively identified, the intruder was a juvenile at the time of the intrusion. Although I'm not convinced that their age is necessarily a good indicator of their skill level.

The author of the programs frequently identified himself/herself, by nickname and/or e-mail address. When the author identified himself/herself, he/she would include various disclaimers, warning that use of the program on a public network was illegal, the author was not responsible for any damage, etc.

The programs were portable across several of the UNIX variants.

System Administrator Lessons Learned

None of the lessons learned here involve anything beyond good basic computer security practices. But the reality of the world we live in is that resource constraints force most system administrators to pick and choose which security measures to implement. Hopefully, this will help system administrators justify more resources and where they do not get more resource allow them to better prioritize their activity.

As noted above most of these systems were compromised long before anyone realized that anything was wrong. Most system administrators became aware of these programs only after authorized users complained about poor network or system performance and the cause of the high network or system utilization was traced back to one of the unauthorized programs. In the only case where the incident was identify by a review of logs, the system administrators allowed the subject to stay on the victim system for an additional week, mostly out of curiosity. They acted on the fact that the system had been compromised and these programs were being installed only after the intruder launched a packet flood/denial of service attack. In addition to the incidents discussed above where the hackers programs were found and analyzed, I am personally familiar with a number of intrusions where the victim became aware of the compromise only after the intruder called him/her, generally in an extortion attempt. This theme of long undiscovered compromises is reinforced by a recent advisory from the National Infrastructure Protection Center (NIPC). The advisory relates to an increase in hacker activity targeting U.S. systems associated with e-commerce. Specifically hackers gaining unauthorized access to these systems and downloading propriety information. The advisory goes on to note that in most cases the activity had been on-going for several months before the victim became aware of the intrusion.

Further, because in each case the intruder had access to the system for a prolonged period of time and multiple programs were involved, the greatest cost was the manpower and down time associated with returning the system to a known safe state.

The lessons learned, or more accurately reinforced, here are:

1. The defense in depth concept must extend beyond the use of multiple tools to prevent the initial intrusion. We need to extend our efforts beyond prevention and detection of the initial intrusion. We should assume that someone will overcome our efforts to keep them out of our system. This means that detecting post compromise unauthorized use. We can do this by looking for activity on port numbers commonly associated with Trojans, backdoor, denial of service tools, and the like. A number of web sites including http://www.simovits.com list these ports. We can and should do this by using Tripwire, or a similar tool, to look for the changes to the system. A complete discussion of Tripwire is beyond the scope of this paper, but for people not familiar with the program, it allows you to do the following: take a "snap shot" of your system, define what to include in that "snap shot," and to find changes by comparing the current "snap shot" to a previous "snap shot." Given the programs we can reasonably expect the intruder to install, I'd suggest that two additional and potentially effective ways to search for unauthorized use are to check the mode of the network card and to look for the presence of a compiler. Very few computer systems have a legitimate reason for running in the promiscuous mode and many systems have no day to day need for a compiler. Neither of these checks is fool proof , however, they are a relatively low effort way to add to your defense in depth.

2. The need for a good system of backups, so you can quickly and confidently return to a pre-compromise state.

3. If we assume that the reason that the intruders are installing sniffers is to capture user ids and passwords, and we know that intruders frequently install a sniffer, this reinforces the need for a good password policy. In particular, if users need to have accounts on systems on both sides of an organizations firewall they need to use two sets of user ids and passwords. Also, if passwords are a consistent target of intruders maybe it makes sense to use a one time password system. *Practical Unix & Internet Security,* by Simson Garfinkel and Gene Spafford, was published in April, 1996. It recommends "Do not require the user to send a reusable password in cleartext over the network connection to authenticate himself. Either use one-time passwords, or some shared, secret method of authentication that does not require sending compromisable information across the network."

Implications for Law Enforcement

The purpose of this section is to help you understand how the functions of unauthorized programs installed can increase the options available to law enforcement, if you choose to notify them. Under current federal law, specifically United States Code Title 18, Section 1030, simply breaking into a computer system and installing an unauthorized program is not in many cases a federal felony. There are exceptions, which typically involve computer systems associated with national security, financial institutions, or medical records. However, the initial reaction to pursuing most intrusions (unauthorized access) involves pursuing charges under Section 1030 section (a)(2) (C) –intentionally accesses a computer without authorization or exceeds authorized access, and thereby obtains information for a protected computer (the term protected computer includes a computer which is used in interstate or foreign commerce or communication) ; or (a)(5)(A and/or B) which cover intentional damage to a protected computer and reckless damage to a protected computer. The exact application of these various sub-sections is not especially important, what is important is that they all require that the intruder cause $5,000.00 or more in damages. In practice this means that someone can exploit a vulnerability, gain root access, and install a variety of programs on your system and frequently, unless the intruder deletes important files, etc., there may not be any legal (federal criminal) consequences. Because even if you firmly identify the intruder, the United States Attorney's Office (the federal prosecutors) in your district may not be able to pursue the case because of the damage requirement. Further, I've found that, even if they do not understand the details of the law, many system administrators understand that it is very difficult to bring charges against a non-destructive intruder. As a result they choose to simply kick out an intruder and do not report the compromise. However, if you look at what the programs left by the intruder actually do, particularly if they resemble what was found in the cases discussed above, there are other sub-sections of Section 1030 that can be used to charge an intruder. If, as is frequently the case, the subject installs a

sniffer/packet capture program, then you may be able to charge the intruder under section 1030 (a)(6). That section deals with trafficking in computer passwords and does not have a damage requirement. In addition, it may be possible to pursue the case using Section 2511, which prohibits the interception of wire, oral, or electronic communications.

Proposed Computer Crime Legislation

A European treaty currently being negotiated by the Council of Europe, and supported by the United States Department of Justice, would make the possession of "hacker tools" illegal. Many in the computer security community have opposed these restrictions fearing that they would interfere with legitimate security work. I believe that some of the common characteristics of the tools found on the victim systems would cause any reasonable professional person in the computer security industry to accept that there are some circumstances under which the possession of certain tools/programs should be criminalized. Clearly some of the tools which might be criminalized under an overly broad treaty or law have legitimate security uses that out weigh any benefit we would derive for criminalizing them. The most obvious example would be tools for scanning a host to determine which ports are open (offering services). However, criminalizing some of the tools commonly installed by these intruders, like programs for launching packet flood attacks, could positively impact the international communities ability to deal with computer crime with out significantly harming legitimate security work. Other programs commonly installed by hackers after a system is compromised, like sniffers/packet capture programs, could be regulated (legal under certain circumstances). For example, only individuals with legitimate administrative responsibility for a computer network could legally possess packet capture tools/sniffers, or tools for cracking passwords. From examining evidence seized in unrelated cases, I know it is not uncommon for individuals who have one or two PCs running Windows 9X, NT, or Linux, and no legitimate access to a computer network (excluding their ISP), to have password cracking programs, sniffer programs, and packet flood programs.

References

1. Simson Garfinkel and Gene Spafford, Practical Unix & Internet Security Second Edition, published April, 1996 by O'Reilly and Associates Inc., Sebastopol, California.

2. Alan Paller, "Fighting Back Against Cybercriminals, Case Studies in Cooperation," May 4, 2000, The SANS Institute http://www.sans.org/newlook/resources/FBACCsld001.htm

3. Steps for Recovering from a UNIX or NT System Compromise, CERT Coordination Center, Carnegie Mellon University, Software Engineering Institute. http://www.cert.org/tech_tips/win-UNIX-system_compromise.html

4. Ports used by Trojans, Simovits Consulting, http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html

5. ONCTek List of possible Trojan/Backdoor port activity, http://www.onctek.com/trojanports.html

6. National Infrastructure Protection Center (NIPC) Advisory 00-60: E-Commerce Vulnerabilities, http://www.nipc.gov

7. Federal Criminal Code and Rules, published by the West Group

8. SANS. "Intrusion Detection Overview and Trends in Internet Attacks" GSEC course materials

Last Updated: May 29th, 2017

Upcoming Training

SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
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 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 Charlotte 2017	Charlotte, NC	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
SANS Cyber Defence Singapore 2017	Singapore, Singapore	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
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
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
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
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 Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event