

Spotting Intrusions: A Real-Life Scenario

Mark Burnett 2000-09-11

Spotting Intrusions: A Real-Life Scenario

by Mark Burnett, Xato Network Security

last updated Mon September 11 2000

Do you ever get that feeling that your web site's security may have been compromised but you do not really know for sure? Sure, you can keep up with patches and make sure all the ACL's are set correctly but with so many new exploits introduced every week and the huge number of exploits that have never been made public, how can you be sure you are protected? Sometimes things just don't feel right and you suspect that you have been attacked. And although some attackers will make it very clear you have been hacked, there are many more who will come and go without you ever noticing, making them the more dangerous of the two.

But how do you know for sure if your security has been compromised? This article will teach you some techniques you can use to detect intrusions as they are happening and give you a leading edge in protecting yourself. Although there are many 3rd party applications that can help with intrusion detection, I am going to only be using built-in Windows 2000 tools to demonstrate the concepts behind detecting intruders. But most importantly, by analyzing how an attacker would behave, we can know which techniques would be most effective in spotting an attack.

Now suppose that for whatever reason, I have determined to penetrate your network. How would I begin? First, I would gather as much information as possible about your network. I would do that through a number of tools such as whois, dig, nslookup, tracert, as well as using a number of publicly available sources of information on the internet. Suppose that through all this research I found a small portion of your network that was not protected by a firewall. I perform a port scan and notice that several machines have ports 135, 139, 389 and 445 open-- a dead giveaway for Windows 2000 boxes. I notice that one of the boxes also has ports 80 and 443 open, which is probably an IIS 5 web server.

So at this point, how could I have been detected? First of all, there was the port scan. During that scan you should have noticed a sudden increase in network traffic. But how can you notice a sudden increase in network traffic? Normally this is where another tool would come in handy but In Windows 2000 that can be done by adding a performance alert for a pre-determined

limit. Good indicators of network traffic would be the TCP-Segments/Sec. or Network Interface-Packets/Sec. Port scans will normally appear as a steady plateau of increased traffic lasting several minutes, depending on how many ports are scanned.

Another simple indicator of network traffic on a computer that is not very busy would be to enable the taskbar icon for your network adaptor. To do that, go to the properties of the network adaptor and check the box for "Show icon in taskbar when connected." With this option selected, an icon will appear in the taskbar that will light up with all incoming and outgoing network traffic.

And finally, there is the built-in command line tool netstat. If you suspect a scan, you can enter the command:

```
Netstat -p tcp -n
```

if you are currently being scanned, and depending on what tool is being used, you may get results something like this:

Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.13.18.201:2572	127.199.34.42:135	TIME_WAIT
TCP	127.13.18.201:2984	127.199.34.42:1027	TIME_WAIT
TCP	127.13.18.201:3106	127.199.34.42:1444	SYN_SENT
TCP	127.13.18.201:3107	127.199.34.42:1445	SYN_SENT
TCP	127.13.18.201:3108	127.199.34.42:1446	SYN_SENT
TCP	127.13.18.201:3109	127.199.34.42:1447	SYN_SENT
TCP	127.13.18.201:3110	127.199.34.42:1448	SYN_SENT
TCP	127.13.18.201:3111	127.199.34.42:1449	SYN_SENT
TCP	127.13.18.201:3112	127.199.34.42:1450	SYN_SENT
TCP	127.13.18.201:3113	127.199.34.42:1451	SYN_SENT
TCP	127.13.18.201:3114	127.199.34.42:1452	SYN_SENT

Notice the sequential ports on both the the local and foreign addresses. Also notice the large number of SYN_SENT entries. Some scanning tools may show ESTABLISHED or TIME_WAIT. The key indicator is the sequence of the ports and the large number of connections from the same host.

But back to the scenario. At this point, I have several paths to go down. One is to look for weak Windows networking passwords and the other is to exploit the web services on the web server I discovered. Since a network logon would give me the most power, I decide to go that route first. I download the account names from one machine and pick one that is rarely used (perhaps a guest account). I attempt several logins to that account until it locks out to determine exactly what the lockout policy is set at. I then put together a script that tries a number of logins for each account (enough to not trigger lockouts). Of course, since the Administrator account doesn't lock out I give it a short list of common passwords to try. I fire up my script and since it might take some time for it to run I also run RFP's Whisker scanner using a script I have customized for IIS servers. I set it to bounce off several public proxy servers then just sit back and wait for results.

In the meantime on your end you should be receiving several alerts from some key intrusion-detection counters. The first one would be your Web Service-Connection Attempts/sec. That would tell you that your web traffic has suddenly increased. Another very important counter would be Web Service-Not Found Errors/sec. Since a web scanner such as Whisker depends upon checking to see if specific urls exist, your performance counters will show a sharp increase in traffic as well as 404 errors. By establishing normal levels beforehand, you can set alerts that would indicate that a cgi scanner is being used against you.

But remember, at the same time there is a brute-force attack going against computers on your network. Again, your performance counters can tip you off. The two counters that will help you are Server-Logon/sec. and Server-Errors Logon. Setting an alert on more than two logons per second and more than five logon errors will quickly let you know if a brute-force attack is underway. A quick check of the Security Event Log will verify that indeed a large number of failed logins were made from one computer.

But by now my scripts have finished and I have found a password. In fact it was a blank administrator password for a system that was just installed yesterday and not totally secured yet. I use that password to connect to that machine and the first thing I do is upload a few tools. Some that come to mind are lsadump2.exe, nc.exe, tlist.exe, and a few of my scanning scripts. I also know that you will have the rest of the tools I need such as nbtstat.exe already on your system. I start the scheduler service and schedule nc.exe to run in one minute and to redirect cmd.exe to a port such as 1234. I wait a minute then use nc.exe locally to connect to the remote command prompt. I run tlist.exe to get a process list (and notice the screen saver is on), run lsadump2.exe to check for stored passwords, then browse through the hard drive

for interesting stuff.

Hopefully, by now you are sitting at that particular system and have brought up Task Manager and noticed cmd.exe with a high process ID. You also notice the scheduler service is running. You take a look at C:\Winnt\SchedLgU.txt and notice a very recent entry for the process nc.exe, which is also in your task list. Using Explorer's find feature, you search for all files created in the last day. Not surprisingly, you see several new executable's in your System32 directory, including nc.exe. You try to End Task the command prompt but are not allowed to end it. At this point an intrusion has been detected and it is time to start collecting evidence. Shutting down the computer at this point would tip off the attacker but you do not want to leave him on long enough to actually take anything. The Event Log shows the failed login attempts and should also show the successful login at the end. However, the event entry does not show the IP address of the computer on the other end, only the computer's name. To determine the IP address on the other end, go to a command prompt and type:

```
netstat -a -n
```

Look for connections to the local TCP ports 139 or 445 and UDP ports 137 and 445.

Save that output to a file and then perform a name lookup on that IP address:

```
nbtstat -A <my IP Address>
```

In your netstat output you should have also noticed a connection to TCP port 1234, which is the netcat session. You may also notice several UDP 137 and 138 connections to other computers in your network.

Meanwhile, I have been scanning other computers on your internal network via my remote netcat command prompt. I have found a share named PUBLIC on a computer named FILESERVER so I map that share and start snooping around.

Because of the NetBIOS connections to other computers in your network, you suspect that this computer is going to be used to compromise the internal network and so at a command prompt you enter the command:

```
net view
```

You notice drive mappings to the internal file server and decide that you have enough evidence and it is about time to pull the plug. You begin to run all the commands again and redirect the output to a file.

After browsing for a bit I run `tlist.exe` again and notice that the screen saver is no longer running and that a command prompt is now open. Not sure if you are on to me, I quickly make a registry entry to run netcat again at startup and then disconnect. I wait about 10 minutes and then ping the computer again and get a Request timed out response. Obviously, you have discovered me so I disconnect my "borrowed" AOL account and start packing my computer.

Now this example is fictional and perhaps a bit simplistic, but it demonstrates many elements of a network-based attack and how to detect such attacks. Nearly all network-based attacks could be detected if you could keep track of the following information:

- Network traffic levels and network connections;
- Web traffic levels and number of pages not found;
- Successful and failed login attempts;
- Changes made to the file system;
- Applications and services currently running;
- Applications that have been scheduled or that will run at startup

By keeping track of those things, a number of hacking attempts can be foiled without using any external intrusion detection software. Of course, other applications can help, but you should always keep in mind each of the six items above.

In the scenario above, a single attack occurred while the administrator was present. In real life there may be a number of attacks occurring 24 hours a day. Some may be simple port scans while others may involve a full compromise of the network. Either way, it will not be realistic to sit and actually watch performance logs all day. But using the lessons learned in that example you can build a system that will work for you. You may have performance alerts sent to you via an e-mail or pager. You may also use the scheduler service to regularly log all currently

running processes or network connections. With a little scripting, scheduling, and a few freeware tools, one could build an intrusion detection system that could far outperform any commercially available intrusion detection software. The key to spotting intruders is not to have more powerful software but to know how intruders work and keeping one step ahead of them.

[Privacy Statement](#)

Copyright 2006, SecurityFocus