Wargames server 1.0 whitepaper

By Auzy

June 07, 2003

The moment the server got samba on it, that really did make it much too easy. Samba is a SMB server, the equivalent of Microsoft file and printer sharing. The version of Samba was 2.2.5, an older version that had a remote root exploit that existed in Samba for the past 8 years.

An exploit script known as sambal.c was used. This script runs in Linux. To compile it, u needed to do "gcc –o sambal sambal.c". And to run "./sambal –b0 IP". Unfortunately, the script believed that the server wasn't samba, while it was known it was, so all instances of:

```
if (is_samba(argv[optind], 2) != 0) {
        fprintf(stderr, "+ Host is not running samba!\n\n");
        return -1;
}
```

had to be removed to force it to think the target was samba.

After re-running it, it worked successfully, and spawned a shell to the remote computer.. since it was /bin/sh, it meant that barely any remote programs functioned correctly because for started, environment variables and paths were non existent.

One method I thought of getting around the environment variables prob was to use VNC (remote X-server), since I could specify the pass for it, I didn't need to know the root users pass, but unfortunately, I discovered (the hard way), that the path variables were used by the program. If I used another VNC server, that could have been fixed maybe (especially if it was open source), but instead I would use another method, I'd go the full way and get shell access.

The easiest way to do this is to simply change the root user's password using passwd, then we could access the server using root: new pass, but that is obtrusive as the root user would lose their access, and it's too obvious.

The 2nd method is to use john the ripper, and copy the information from /etc/shadow (which can be seen using vi /etc/shadow as more and less and pico were screwed.
"Root:4z5Oy/QzpULE6:12209:0:10000::::" was the data needing to be decrypted using john the ripper, but it was a long pass and thus would take too long.

Finally, it was determined the best way was to make my own user. adduser and useradd commands didn't work because there was no path.. so instead, I went
cd /etc
echo "auzy:x:0:0:root:/root:/bin/bash" >> passwd
echo "auzy:!:12209:0:10000::::" >> shadow

vi couldn't be used easily. For starters, the arrow keys couldn't be used (j and l must be), and u had to visualise at the stuff on the screen to know what u were doing). But it could work. Pico wouldn't start because environment variables were missing.

Doing this procedure effectively created the user auzy, with home directory /root, root privileges and no pass.

There were 2 methods of shell access on the server, telnet and SSH. Telnet is unencrypted and thus ssh was a much better choice. To access it, you need a SSH client from www.ssh.com. However, it would not let me log in as auzy:blank. The reason was due to protection built into the SSH server that disallowed logging in as blank pass users, so in the sambal shell, I changed user to auzy by using "su auzy", which switched user to auzy, and typed passwd, entered a pass and now SSH access worked. I couldn't put the pass directly into shadow because I didn't know the encryption codes, and therefore

it wasn't possible, but passwd automatically encrypted the pass and therefore it was possible to produce a pass that way.

SSH access was much better as everything worked.. To modify the page, it was simply a matter of
cd /srv/www/htdocs
vi index.php
:wq (write not permitted)..
Writing wasn't permitted, so it was harder, while I am aware :wq! Overrides the write/read access, I didn't before, so before I did
chmod +w index.php
modified the file
chmod –w index.php

To upload the Britney spears midi to the server, wget URL was used, which downloaded a file to that server.

And I was finally done.

Finally, I cleared the .bash_history in /root to make it harder for people joining the server.

After woods, after I've had some fun, I would also inevitably remove my username/password too from /etc/passwd and /etc/shadow using vi.

The addition of samba to the server made it pretty much a script kiddie operation(especially after someone +w'ed index.php and forgot to change it back in later hacks), making it easier for those people joining, so I changed it back to make it in the same state it was originally.. And to make it slightly harder.

One funny thing I noticed was some people decided that they would remove the root pass on the server... What they didn't realise is that that would have been useless, because SSH wouldn't let them login anyway, for trying to use a blank pass. so I just sat there laughing, because they didn't know what they were doing, and entertaining myself at peoples bash_history logs as they tried all kinds of wacky commands that couldn't work, at all.