Session Four – Wireshark Practical

This session, we are going to put our packet capture analysis skills to the test by solving a real-life scenario.

An adversary has managed to download and run a piece of malware on one of your critical servers. Your boss is furious and wants to know how the attack was able to happen and what the attacker was able to do. The only thing you have to go off is a packet capture of the attack, captured by your automated logging system.

Each main part of the puzzle is on a new page to prevent spoliers from reading ahead.

The follow stream tool in Wireshark is key here to being able to read each TCP session at a glance.

Part 0 – Installing Wireshark (if not here last week)

- The WireShark tool can be downloaded for all major OSs from www.wireshark.org/download.html -it is GUI based and may prompt you to download a couple of additional drivers.

Part 1 – Getting the context

- Which IP addresses were involved? What are the attackers' and victim's IPs? Where is the attacker geographically located?
  - Attacker: 98.114.205.102, Victim: 192.150.11.111
  - Attacker located in USA, probably in Philidelphia
- How many TCP sessions are there in total within the packet capture?
  - 5 Total sessions, of which four are attack related.

Part 2 – Solving the crime

By looking at the individual TCP sessions in order we can piece together the stages of the attack and work out how the attacker was able to download and execute malware on the system. Try to work out what transpired and come up with a step by step analysis of how the attacker gained access, downloaded malware and then executed it. If you need further guidance, the following sections break this down into manageable chunks.

General overview: Attacker took advantage of vulnerable DSSETUP protocol, which had not been patched, to execute code remotely on the Windows 2000 server. The attacker made the server connect to a FTP server, download Malware and run it. This malware is likely to have connected the server to a botnet through a backdoor.

Part 2a – Initial Entry

The attacker used a vulnerable Windows protocol to gain access to the system. This can be found in the TCP session starting at packet #5.

- What network-based program is our vulnerable server running?
  - Server Message Block (SMB)
- What operating system is it using?
  - Windows 2000
- What vulnerable protocol is used by the attacker?
  - Hint: Try googling the protocols Wireshark highlights for that TCP session. Do any have any major known vulnerabilities?
  - DSSETUP
- What would that vulnerability let the attacker do?
  - Execute code remotely on the server via a buffer overflow

Part 2b – Code execution

Now the attacker has gained a way to execute code on the vulnerable server. Look for other TCP sessions which may contain the commands input to the server.

- The code executed makes several references to 'o'. What does 'o' refer to, and how is it used?
    - This is a file used to store FTP commands, which are then used in conjunction with FTP to connect to the malicious FTP server. The file is then deleted afterwards to hide the evidence.
- What is the name of the malware downloaded? Where is it downloaded from?
    - Malware downloaded is ssms.exe it is downloaded over port 8884

Part 2c – Downloading the malware

2b Highlighted that the attacker executed code on the vulnerable server to make it connect to a ftp server and download malware. From looking at the TCP session between the vicitim and the attackers' FTP server we can learn more about what happened.

- What is the IP of the attackers' ftp server? Is this the same as anything else? Is that good offensive cyber practice?
    - It is the same as the main attacker IP. This is bad tradecraft as it makes it easier to link the attack back to the attacker.
- What is the name of the ftp server used?
    - NzmxFtpd
- Looking at the packets sent by the ftp server, what is the likely type of payload on the malware downloaded?
    - Likely a rootkit (gain root access to server). Likely provides attacker with remote access to machine so that it can be used in a botnet etc.

Extension – Part 3 Malware Analysis

If you have got this far, you have successfully solved the question of how the attacker was able to download and run malware on the vulnerable server. The ftp server gives us a good idea of what the Malware might do, but we can also do our own work to try and find out more, as the final TCP session provides the raw code of the malware itself (as it was downloaded from the FTP server to the victim).

- Take a hash of the malware. Does this match any known signatures?
- Use a tool such as strings to pull out any strings from the raw malware. Do these tell us anything useful?