Scan of the Month Challenge #34

Brief Official Write-Up by Anton Chuvakin

Additional setup info

- The victim server ('combo') runs multiple virtual IP addresses in the vicinity of 11.11.79.67. Specific virtual IP addresses can be seen in the syslog ('messages' file) during the reboot on Feb 11.
- The 11.11.79.64 system was a router, that connected the honeynet network to the outside
- The 11.11.79.65 system was a fake target for remote syslog. It does not really receive the logs, they are sniffed by a monitoring system instead.
- Neither 'bridge' nor 'bastion' had IP addresses on the honeynet network. 'Bastion' was running in "stealth" sniffing mode and 'bridge' was using Layer 2 filtering.

Questions and Answers

Q1: What are the significant events that happened on the honeypot in the time period covered by the logs? Show how you analyzed the data to paint the picture of those events.
A1: Significant benign events:

- Reboot: Feb 11, 2005
- AWSTATS installed: Feb 25, 2005

Significant malicious events:

- Compromise thru AWSTATS (1): Feb 26
- Compromise thru AWSTATS (2): Mar 4
- Successful login via SSH (user 'test'): Mar 6,13

The data was analyzed in daily batches in the beginning each day, using nFX OSP software.

Q2: Was the system compromised? How do you know? If yes, how many times and by how many attackers? What would you consider the most compelling evidence of the compromise available if you find that the system was indeed compromised?
A: The system was indeed compromised at least twice. Here we define "compromise" as a successful attack, followed by the some follow-up activity.
The most compelling evidence of compromise (in our opinion, as honeypot owners) was the outbound IRC communication. While 'wget' and 'lynx' malicious software download attempts provide fairly compelling evidence, IRC implies that the intrusion succeeded, the attacker has some degree of control over the machine and that he managed to install his/her own software (an IRC client or bot). In this case, we observe an abundance of IRC activity (channel joins, nick changes, operator activity, multiple TCP 6667 connections, etc) detected by Snort and reported by iptables

Here are some of the IRC traces from the snort logs:

- Feb 26 19:01:10 bastion snort: [1:542:11] CHAT IRC nick change [Classification: Potential Corporate Privacy Violation] [Priority: 1]: {TCP} 11.11.79.67:1060 -> 193.110.95.1:6667
- Feb 26 19:01:11 bastion snort: [1:1463:6] CHAT IRC message [Classification: Potential Corporate Privacy Violation] [Priority: 1]: {TCP} 193.110.95.1:6667 -> 11.11.79.67:1060
- Feb 27 00:40:10 bastion snort: [1:2000348:3] BLEEDING-EDGE IRC - Channel JOIN on non-std port [Classification: A Network Trojan was detected] [Priority: 1]: {TCP} 11.11.79.67:1755 -> 66.198.160.2:8888
- Feb 27 02:45:39 bastion snort: [1:2000345:3] BLEEDING-EDGE IRC - Nick change on non-std port [Classification: A Network Trojan was detected] [Priority: 1]: {TCP} 11.11.79.67:1061 -> 66.198.160.2:8888

and many others, all coming from our compromised system. Those occurred on Feb 26 (started) and then continued in March. It appears that later the attacker has abandoned the system. We cannot exclude the possibility that he was spooked by the limitation of outbound connectivity.

Q3: If this were the evidence from a production system, how would you learn that the machine was compromised, given the data available? For this question, assume you do not have the honeynet-specific data streams, such as sebek2 or bash logger, just like in this challenge.
A: Even for production environment, outbound IRC initiated by a web server is a 100% reliable indication that the

successful attack has taken place (IDS logs needed). We cannot reliably conclude a compromise from syslog messages. We can make an educated guess that a machine was compromised from Apache logs (access_log) since it responded with a code 200 to an 'awstat.pl' attack request (web server logs). Iptables log indicating outbound connection from a server can also provide basis for a fairly educated guess (iptables logs).

Q4: What else was going on at the system at the same time? What times of "Internet noise" can you categorize, given the data? Is there anything out of the ordinary with the noise levels? What attack and probe types observed actually had a chance of affecting the target?
A: We can briefly summarize the types of noise by Snort alarm or by destination port (left as an exercise to the reader).

- **ICMP Destination Unreachable (Port Unreachable) and other ICMP alerts:** these ICMP may be generated due to variety of reasons (worms, DDoS backscatter, etc)
- **MS-SQL Worm propagation attempt:** our good old friend, Mr Slammer. Everybody knows this one.
- **WEB-MISC WebDAV search access, WEB-IIS cmd.exe access, WEB-FRONTPAGE /_vti_bin/ access and other IIS attacks:** lots of attacks target IIS version 4 and 5. Will IIS6 be different?
- **POLICY SMTP relaying denied:** a scourge of 2000s, folks scan for relay and try to abuse them right away
- **Typot trojan traffic:** this is a weird one. It is likely a "false positive", but I am not quite sure why it is being generated.
- **WEB-MISC bad HTTP/1.1 request:** this might well be Apache's Slapper trace or similar activity. Some IIS attacks are also not unlikely.
- **RPC portmap status request UDP:** some folks still kind forget the good ole 2000, when Linux was hackable with a RPC exploit in older RedHat and other distros
- **SCAN SSH Version map attempt:** SSH login guessing appeared out of woodwork some time ago and is still ongoing. What is weird about it is that they rarely follow up even after successful password guess...
- **SCAN NMAP -sA** : looks like somebody out there has discovered the pleasures of 'nmap' (ACK scan - oooh, how advanced!)
- **Various HTTP attempts** : not all HTTP accesses inn the logs fit under IIS attacks, there are other attempts to for pages and scripts (some more mysterious than others)

In addition, Messenger spam (ports 135-139) is still rampant. The port profile closely matches what is observed by DShield.org and other sources.

Q5: Do you think that the time was synchronized between the various monitoring systems (where Snort and iptables logs were collected) and a victim system(where syslog and Apache logs were collected)?
A: 1. The time between the monitoring box ('bastion') and the gateway ('bridge') was synchronized thru NTP. Thus, iptables and Snort records are in sync.
2. On the other hand, the time between those systems above and the victim server ('combo') was not synchronized. In fact, there was a huge gap of 4:47 hours, due to the carelessness of the honeypot operator (i.e. me :-)) However, such situation is very common is production environments and thus provided a fun challenge for the participants. It is not easy to keep the honeynet time-synchronized with the analysis systems, since no direct communication is established. However, the honeynet was later configured to talk to a global time server and so were the analysis systems (independently).

Q6: Describe the procedures and tools of that you used to analyze all the distinct log sources together.
A: Logs were all imported into netForensics nFX OSP SIM software. The system converted them to the uniform format and the built-in tools were then used to summarize, visualize and correlate logs as well as track attacker's activities.

Conclusion Logs from multiple sources (host and network), combined together, provide tremendous value during the compromised system analysis and allow a complete incident reproduction even without using honeypot specific tools such as Sebek

---

Created by Anton Chuvakin, Ph.D., GCIA, GCIH, GCFA

Last modified: Fri May 20 00:51:01 Eastern Standard Time 2005