**Holiday Challenge – Dynamic Stack Generation**

**Author: Ashutosh Gautam(JumpThere)**

Section: Binary

Upon unpacking the docker, we are presented with a binary file which isn’t stripped – meaning all symbols and linking are preserved.

When executed the ‘readelf’ command against the binary, we see the below showing a potential IoC – an indicator of compromise. We aren’t sure as to the capability it has now, but we will see how it fares along the analysis journey.

1. ‘readelf’ against the target binary below:

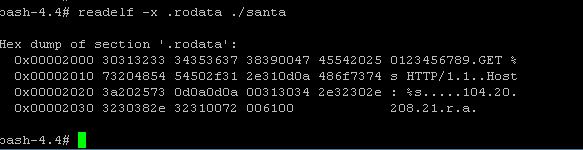


Fig 1.1 – readelf showing the .rodata section holding a potential Indicator of Compromise. This doesn’t guarantee it is a real one.

1. We then proceed to see what other capability the binary beholds. We have few options here, one easy way to spot the obvious is to use ‘strings’ command, or use ‘radare2’ package installed on the docker image.

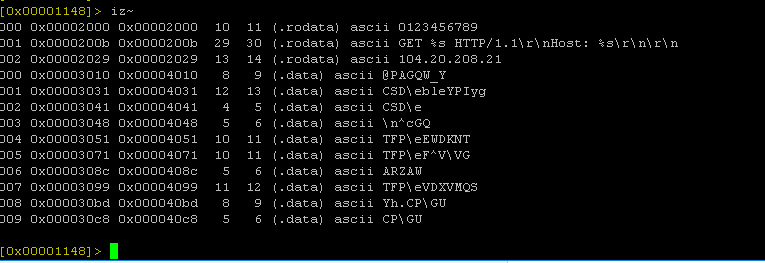


Fig: 1.2 – radare’s function to snoop strings from section data.

1. We then proceed to further analyze the binary inside ‘radare2’ which would show us the list of imports the binary has.

root@container# radare2 ./santa

# Analyze all

# aaaa

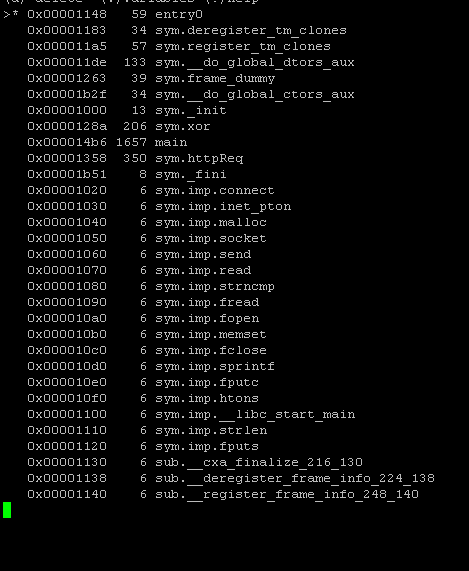


Fig 1.3 – Third column houses list of imports the binary has. ‘Sym.imp’ means the binary had list of exports not necessarily present in the binary by default.

1. Up until now, only the static analysis was done, we then proceed to see how the binary behaves upon execution.
2. Once executed, we then move to host and see if the binary would create a network connection outbound:

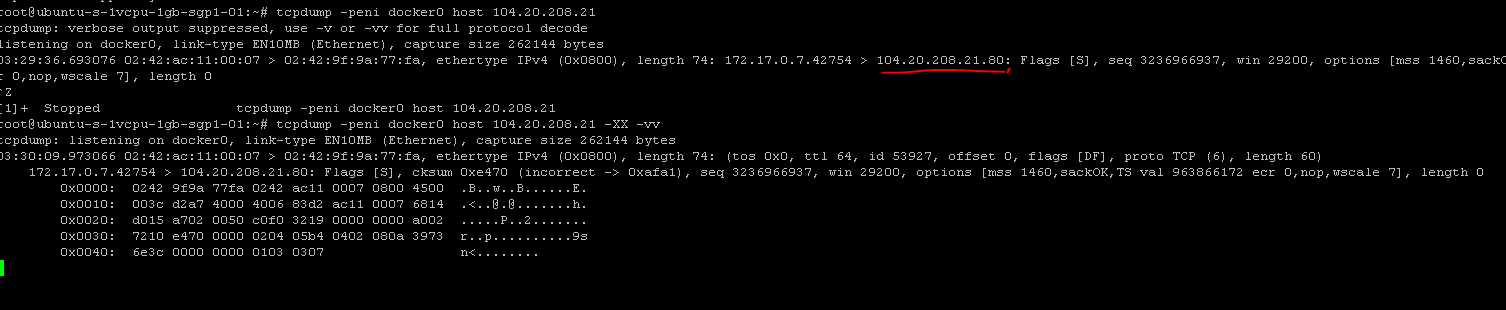


Fig 1.4: binary santa initiating an outbound connection to 104.20.208.21 at port 80.

🡺 Connection to PasteBin:

- While the real connection never happens, there is a reference to active PasteBin entries(2 of them found during the time I took reversing the binary) – <http://pastebin.com/raw/WZRaiyHU>

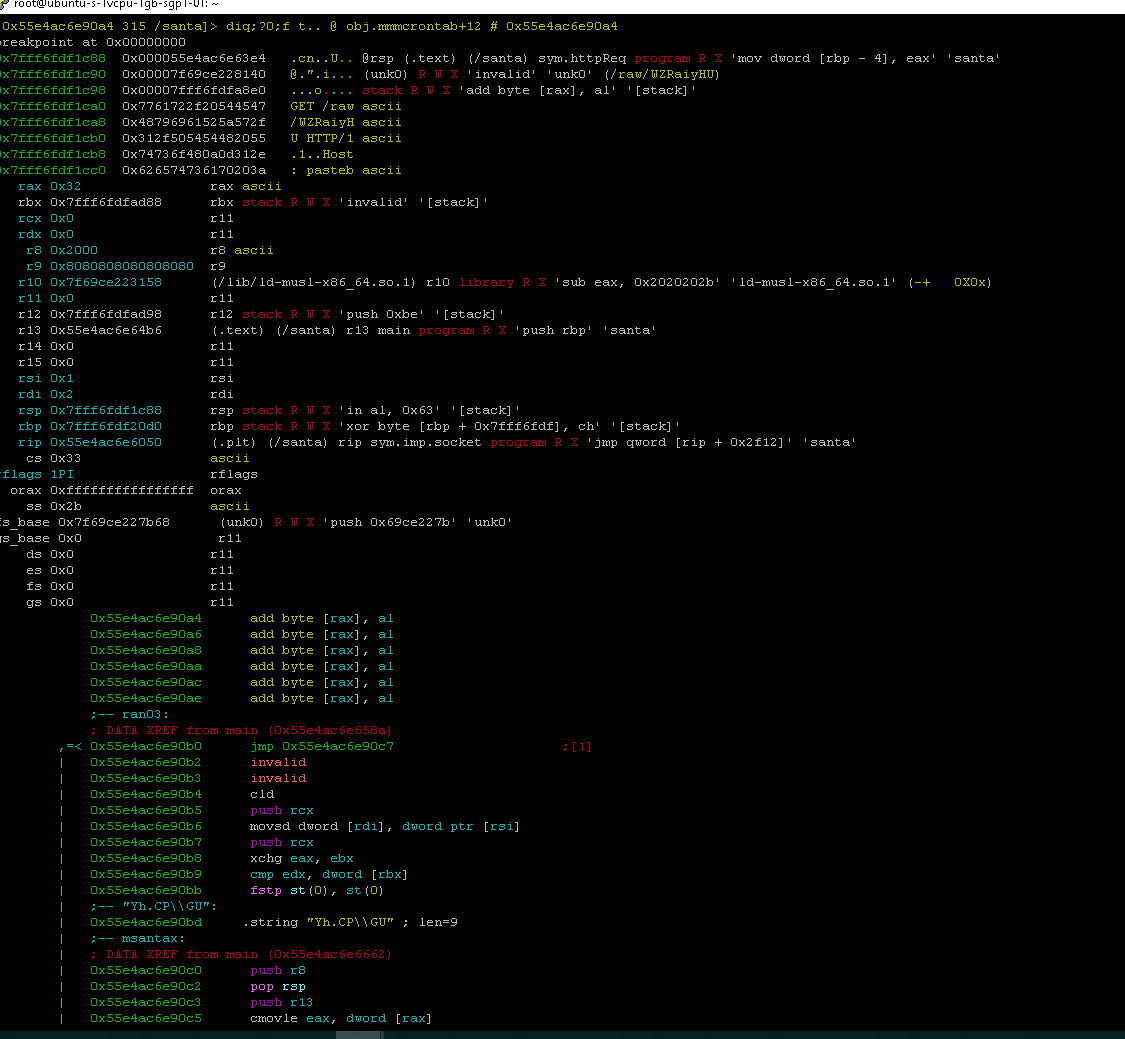


Fig 1.5 – Stack containing artifacts to pastebin – This in turn yields below payload:

santa:x:1337:1337:,,,:/home/santa:/bin/bash

santa:$6$8nc.g49B$d7mLPa0jb9zjrFnTJJm4TA98bls5NXNlaLsZ6D.QFxGopK1/8nL8JXKGJgqRh0OvIIjONng3O5xMy2WIQvgjP.:17886:0:99999:7:::

🡺Connection 2:

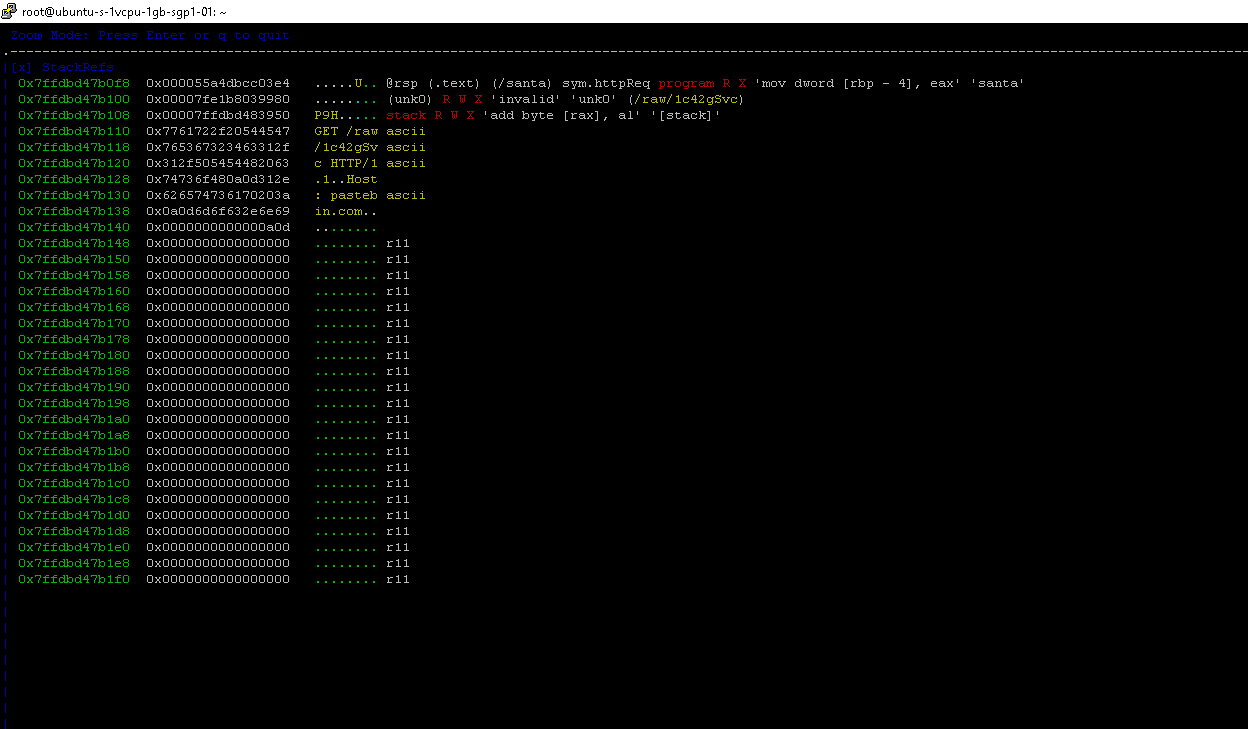


Fig 1.6 – Stack referencing other pastebin entries – Payload below:

\*/5 \* \* \* \* santa /usr/bin/logger "Merry Xmas!"

**Conclusion:**

While the binary isn’t malicious in nature, it references one HTTP call at port 80, and 2 pastebin entries. These are enough evidence to have end-users/firewall/SIEM devices to warn/notify of an intrusion.

**Total IoCs found – 3**

1. <http://pastebin.com/raw/1c42gSvc>
2. <http://pastebin.com/raw/WZRaiyHU>
3. 104.20.208.21: 80, possibly benign.

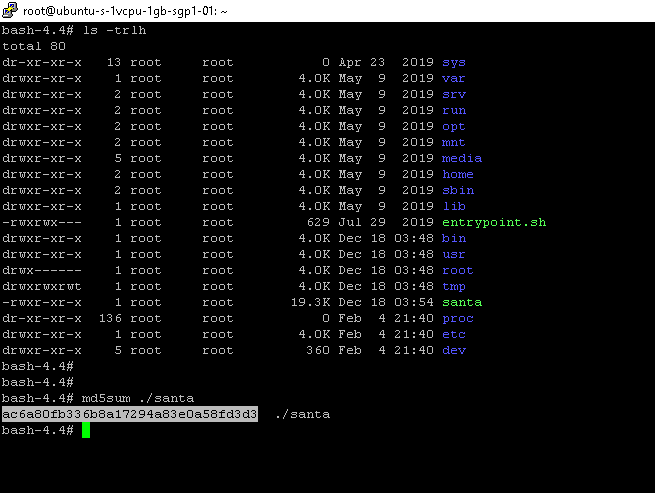
Tools used:

Radare2, rabin, rax, tcpdump, r2

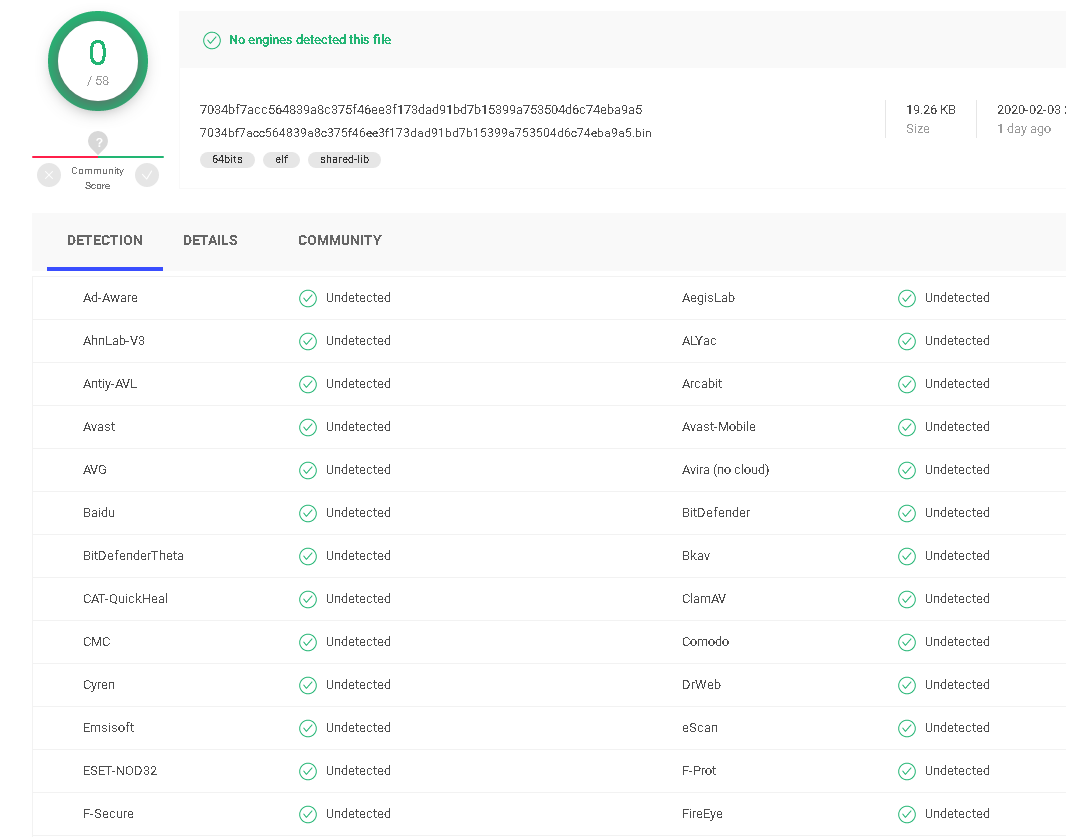
EoF –

Red Team Handlers – you are welcome to chuck this through to a sandbox/VM. As this sample is dynamically generated, a sandbox wouldn’t be able to catch it thru.

Evidence thru VirusTotal – as of 2/5/2020 – against the hash: ac6a80fb336b8a17294a83e0a58fd3d3



Snippet capturing a md5 against the target binary



Snippet capturing 0 hits against the malware

Link to virustotal: <https://www.virustotal.com/gui/file/7034bf7acc564839a8c375f46ee3f173dad91bd7b15399a753504d6c74eba9a5/detection>