Hunting Evil with YARA (Linux Edition)

Let's face the reality of cybersecurity operations: often, as Security Analysts, we don't get the luxury of direct access to a potentially compromised system. Imagine, we've got suspicious flags waving from a remote machine, but due to organizational boundaries, permissions, or logistical issues, we just can't lay our hands on it. It feels like knowing there's a potential fire but not being able to see or touch it directly. This situation can be nerve-wracking, but it's a challenge we've learned to tackle.

Here's where things get interesting. Even if we can't access the machine, in many cases, a memory capture (or memory dump) from the suspicious system can be handed over to us in the Security Operations Center (SOC). It's akin to receiving a snapshot of everything happening in the system at a particular moment. And just because we have this snapshot, doesn't mean our hands are tied.

Luckily, our trusty tool YARA comes to the rescue. We can run YARA-based scans directly on these memory images. It's like having x-ray vision: we can peer into the state of the system, looking for signs of malicious activity or compromised indicators, all without ever having direct access to the machine itself. This capability not only enhances our investigative prowess but also ensures that even remote, inaccessible systems don't remain black boxes to us. So, while the direct path may be blocked, with tools like YARA and our expertise, we always find a way to shine a light into the shadows.

Hunting for Evil Within Memory Images with YARA

Incorporating YARA extends the capabilities of memory forensics, a pivotal technique in malware analysis and incident response. It equips us to traverse memory content, hunting for telltale signs or compromise indicators.

YARA's memory image scanning mirrors its disk-based counterpart. Let's map out the process:

- **Create YARA Rules**: Either develop bespoke YARA rules or lean on existing ones that target memory-based malware traits or dubious behaviors.
- **Compile YARA Rules**: Compile the YARA rules into a binary format using the `yarac` tool (YARA Compiler). This step creates a file containing the compiled YARA rules with a `.yrc` extension. This step is optional, as we can use the normal rules in text format as well. While it is possible to use YARA in its human-readable format, compiling the rules is a best practice when deploying YARA-based detection systems or working with a large number of rules to ensure optimal performance and effectiveness. Also, compiling rules provides some level of protection by converting them into binary format, making it harder for others to view the actual rule content.
- **Obtain Memory Image**: Capture a memory image using tools such as DumpIt, MemDump, Belkasoft RAM Capturer, Magnet RAM Capture, FTK Imager, and LiME (Linux Memory Extractor).
- **Memory Image Scanning with YARA**: Use the `yara` tool and the compiled YARA rules to scan the memory image for possible matches.

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, let's SSH into the Target IP using the provided credentials. The vast majority of the actions/commands covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

For instance, we have a memory snapshot named `compromised_system.raw` (residing in the `/home/htb-student/MemoryDumps` directory of this section's target) originating from a system under the siege of `WannaCry` ransomware. Let's confront this image with the `wannacry_artifacts_memory.yar` YARA rule (residing in the

? Go to Questions

Table of Contents

Introduction to YARA & Sigma

Leveraging YARA

YARA and YARA Rules

Developing YARA Rules

Hunting Evil with YARA (Windows Edition)

Hunting Evil with YARA (Linux Edition)

Hunting Evil with YARA (Web Edition)

Leveraging Sigma

Sigma and Sigma Rules

Developing Sigma Rules

Hunting Evil with Sigma (Chainsaw Edition)

Hunting Evil with Sigma (Splunk Edition)

Skills Assessment

Skills Assessment

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

/home/htb-student/Rules/yara directory of this section's target).

Here's an example command for YARA-based memory scanning:

Hunting Evil with YARA (Linux Edition)

```
MisaelMacias@htb[/htb]$ yara /home/htb-student/Rules/yara/wannacry_artifacts_memory.yar /home/htb-s
Ransomware_WannaCry /home/htb-student/MemoryDumps/compromised_system.raw
0x4e140:$wannacry_payload_str1: tasksche.exe
0x1cb9b24:$wannacry_payload_str1: tasksche.exe
0xdb564d8:$wannacry_payload_str1: tasksche.exe
0x13bac36c:$wannacry_payload_str1: tasksche.exe
0x16a2ae44:$wannacry_payload_str1: tasksche.exe
0x16ce55d8:$wannacry_payload_str1: tasksche.exe
0x17bf1fe6:$wannacry_payload_str1: tasksche.exe
0x17cb8002:$wannacry_payload_str1: tasksche.exe
0x17cb80d0:$wannacry_payload_str1: tasksche.exe
0x17cb80f8:$wannacry_payload_str1: tasksche.exe
0x18a68f50:$wannacry_payload_str1: tasksche.exe
0x18a9b4b8:$wannacry_payload_str1: tasksche.exe
0x18dc15a8:$wannacry_payload_str1: tasksche.exe
0x18df37d0:$wannacry_payload_str1: tasksche.exe
0x19a4b522:$wannacry_payload_str1: tasksche.exe
0x1aac0600:$wannacry_payload_str1: tasksche.exe
0x1c07ed9a:$wannacry_payload_str1: tasksche.exe
0x1c59cd32:$wannacry_payload_str1: tasksche.exe
0x1d1593f0:$wannacry_payload_str1: tasksche.exe
0x1d1c6fe2:$wannacry_payload_str1: tasksche.exe
0x1d92632a:$wannacry_payload_str1: tasksche.exe
0x1dd65c34:$wannacry_payload_str1: tasksche.exe
0x1e607a1e:$wannacry_payload_str1: tasksche.exe
0x1e607dca:$wannacry_payload_str1: tasksche.exe
0x13bac3d7:$wannacry_payload_str2: www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x197ba5e0:$wannacry_payload_str2: www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x1a07cedf:$wannacry_payload_str2: www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x1a2cb300:$wannacry_payload_str2: www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x1b644cd8:$wannacry_payload_str2: www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x1d15945b:$wannacry_payload_str2: www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x1dd65c9f:$wannacry_payload_str2: www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
0x450b048:$wannacry_payload_str3: mssecsvc.exe
0x5a7f3d4:$wannacry_payload_str3: mssecsvc.exe
0xda1c350:$wannacry_payload_str3: mssecsvc.exe
0x12481048:$wannacry_payload_str3: mssecsvc.exe
0x17027910:$wannacry_payload_str3: mssecsvc.exe
0x17f0dc18:$wannacry_payload_str3: mssecsvc.exe
0x18c360cc:$wannacry_payload_str3: mssecsvc.exe
0x1a2a02f0:$wannacry_payload_str3: mssecsvc.exe
0x13945408:$wannacry_payload_str4: diskpart.exe
0x19a28480:$wannacry_payload_str4: diskpart.exe
```

Beyond standalone tools, diving deeper into memory forensics offers a plethora of avenues. Integrating YARA within memory forensics frameworks amplifies its potential. With the Volatility framework and YARA operating in tandem, WannaCry-specific IOCs can be detected seamlessly.

The Volatility framework is a powerful open-source memory forensics tool used to analyze memory images from various operating systems. YARA can be integrated into the Volatility framework as a plugin called `yarascan` allowing for the application of YARA rules to memory analysis.

The Volatility framework is covered in detail inside HTB Academy's `Introduction to Digital Forensics` module.

For now, let's only discuss how YARA can be used as a plugin in the Volatility framework.

Single Pattern YARA Scanning Against a Memory Image

In this case, we'll specify a YARA rule pattern directly in the command-line which is searched within the memory image by the `yarascan` plugin of Volatility. The string should be enclosed in quotes (`"`) after the `-U` option. This is useful when we have a specific YARA rule or pattern that we want to apply without creating a separate YARA rules file.

From previous analysis we know that WannaCry malware attempt to connect to the following hard-coded URI `www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com`

Introducing this pattern within the command line using `-U "www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"`

prompts a search within the `compromised_system.raw` memory image.

Hunting Evil with YARA (Linux Edition)

```
MisaelMacias@htb[/htb]$ vol.py -f /home/htb-student/MemoryDumps/compromised_system.raw yarascan -U
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12: Cry
  from cryptography.hazmat.backends.openssl import backend
Rule: r1
Owner: Process svchost.exe Pid 1576
0x004313d7  77 77 77 2e 69 75 71 65 72 66 73 6f 64 70 39 69   www.iuqerfsodp9i
0x004313e7  66 6a 61 70 6f 73 64 66 6a 68 67 6f 73 75 72 69   fjaposdfjhgosuri
0x004313f7  6a 66 61 65 77 72 77 65 72 67 77 65 61 2e 63 6f   jfaewrwergwea.co
0x00431407  6d 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00   m...............
0x00431417  00 f0 5d 17 00 ff ff ff ff 00 00 00 00 00 00 00   ..].............
0x00431427  00 00 00 00 00 00 00 00 00 20 00 00 00 04 00 00   ................
0x00431437  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431447  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431457  00 00 00 00 00 00 50 51 17 00 00 00 00 00 00 00   .....PQ.........
0x00431467  00 13 00 00 00 b8 43 03 00 00 00 00 00 00 00 00   ......C.........
0x00431477  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431487  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431497  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x004314a7  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x004314b7  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x004314c7  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
Rule: r1
Owner: Process svchost.exe Pid 1576
0x0013dcd8  77 77 77 2e 69 75 71 65 72 66 73 6f 64 70 39 69   www.iuqerfsodp9i
0x0013dce8  66 6a 61 70 6f 73 64 66 6a 68 67 6f 73 75 72 69   fjaposdfjhgosuri
0x0013dcf8  6a 66 61 65 77 72 77 65 72 67 77 65 61 2e 63 6f   jfaewrwergwea.co
0x0013dd08  6d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   m...............
---SNIP---
```

This option allows us to directly specify a YARA rule string within the command-line itself. Let's see how we can search for the content of a whole YARA rule file (i.e. `.yar` rule file) in memory image files.

Multiple YARA Rule Scanning Against a Memory Image

When we have multiple YARA rules or a set of complex rules that we want to apply to a memory image, we can use the `-y` option followed by the rule file path in the Volatility framework, which allows us to specify the path to a YARA rules file. The YARA rules file (`wannacry_artifacts_memory.yar` in our case) should contain one or more YARA rules in a separate file.

The YARA rules file we will use for demostration purposes is the following.

Hunting Evil with YARA (Linux Edition)

```
MisaelMacias@htb[/htb]$ cat /home/htb-student/Rules/yara/wannacry_artifacts_memory.yar
rule Ransomware_WannaCry {

    meta:
        author = "Madhukar Raina"
        version = "1.1"
        description = "Simple rule to detect strings from WannaCry ransomware"
        reference = "https://www.virustotal.com/gui/file/ed01ebfbc9eb5bbea545af4d01bf5f107166184048


    strings:
        $wannacry_payload_str1 = "tasksche.exe" fullword ascii
        $wannacry_payload_str2 = "www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" ascii
        $wannacry_payload_str3 = "mssecsvc.exe" fullword ascii
        $wannacry_payload_str4 = "diskpart.exe" fullword ascii
        $wannacry_payload_str5 = "lhdfrgui.exe" fullword ascii

    condition:
        3 of them
```

Let's run Volatility with the rule `wannacry_artifacts_memory.yar` (residing in the `/home/htb-student/Rules/yara` directory) to scan the memory image `compromised_system.raw` (residing in the `/home/htb-student/MemoryDumps` directory)

Hunting Evil with YARA (Linux Edition)

```
MisaelMacias@htb[/htb]$ vol.py -f /home/htb-student/MemoryDumps/compromised_system.raw yarascan -y
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12: Cry
  from cryptography.hazmat.backends.openssl import backend
Rule: Ransomware_WannaCry
Owner: Process svchost.exe Pid 1576
0x0043136c  74 61 73 6b 73 63 68 65 2e 65 78 65 00 00 00 00   tasksche.exe....
0x0043137c  52 00 00 00 43 6c 6f 73 65 48 61 6e 64 6c 65 00   R...CloseHandle.
0x0043138c  57 72 69 74 65 46 69 6c 65 00 00 00 43 72 65 61   WriteFile...Crea
0x0043139c  74 65 46 69 6c 65 41 00 43 72 65 61 74 65 50 72   teFileA.CreatePr
0x004313ac  6f 63 65 73 73 41 00 00 6b 00 65 00 72 00 6e 00   ocessA..k.e.r.n.
0x004313bc  65 00 6c 00 33 00 32 00 2e 00 64 00 6c 00 6c 00   e.l.3.2...d.l.l.
0x004313cc  00 00 00 00 68 74 74 70 3a 2f 2f 77 77 77 2e 69   ....http://www.i
0x004313dc  75 71 65 72 66 73 6f 64 70 39 69 66 6a 61 70 6f   uqerfsodp9ifjapo
0x004313ec  73 64 66 6a 68 67 6f 73 75 72 69 6a 66 61 65 77   sdfjhgosurijfaew
0x004313fc  72 77 65 72 67 77 65 61 2e 63 6f 6d 00 00 00 00   rwergwea.com....
0x0043140c  00 00 00 00 01 00 00 00 00 00 00 00 f0 5d 17 00   .............]..
0x0043141c  ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0043142c  00 00 00 00 20 00 00 00 04 00 00 00 01 00 00 00   ................
0x0043143c  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0043144c  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0043145c  50 51 17 00 00 00 00 00 00 00 00 13 00 00 00 00   PQ..............
Rule: Ransomware_WannaCry
Owner: Process svchost.exe Pid 1576
0x004313d7  77 77 77 2e 69 75 71 65 72 66 73 6f 64 70 39 69   www.iuqerfsodp9i
0x004313e7  66 6a 61 70 6f 73 64 66 6a 68 67 6f 73 75 72 69   fjaposdfjhgosuri
0x004313f7  6a 66 61 65 77 72 77 65 72 67 77 65 61 2e 63 6f   jfaewrwergwea.co
0x00431407  6d 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00   m...............
0x00431417  00 f0 5d 17 00 ff ff ff ff 00 00 00 00 00 00 00   ..].............
0x00431427  00 00 00 00 00 00 00 00 20 00 00 00 04 00 00 00   ................
0x00431437  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431447  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431457  00 00 00 00 00 50 51 17 00 00 00 00 00 00 00 00   .....PQ.........
0x00431467  00 13 00 00 00 b8 43 03 00 00 00 00 00 00 00 00   ......C.........
0x00431477  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431487  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00431497  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x004314a7  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x004314b7  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x004314c7  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
Rule: Ransomware_WannaCry
Owner: Process svchost.exe Pid 1576
0x0040e048  6d 73 73 65 63 73 76 63 2e 65 78 65 00 00 00 00   mssecsvc.exe....
0x0040e058  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e068  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e078  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e088  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e098  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e0a8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e0b8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e0c8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e0d8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e0e8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e0f8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e108  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e118  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e128  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0040e138  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
Rule: Ransomware_WannaCry
Owner: Process svchost.exe Pid 1576
---SNIP---
```

We can see in the results that the `yarascan` plugin in Volatility is able to find the process `svchost.exe` with PID `1576` in the memory image of the compromised system.

In summary, the `-U` option allows us to directly specify a YARA rule string within the command-line, while the `-y` option is used to specify the path to a file containing one or more YARA rules. The choice between the two options depends on our specific requirements and whether we have a single rule or a set of rules to apply during the analysis.

PROTOCOL

UDP 1337 TCP 443

DOWNLOAD VPN CONNECTION FILE

Connect to Pwnbox

Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK 160ms

Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

Download VPN Connection File

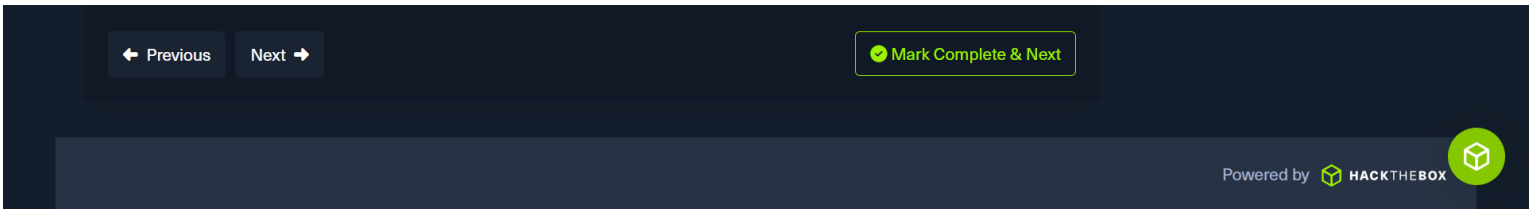
Target(s): Click here to spawn the target system!

SSH to with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 2 Study the following resource https://blogs.vmware.com/security/2022/09/threat-report-illuminating-volume-shadow-deletion.html to learn how WannaCry performs shadow volume deletion. Then, use yarascan when analyzing "/home/htb-student/MemoryDumps/compromised_system.raw" to identify the process responsible for deleting shadows. Enter the name of the process as your answer.

@WanaDecryptor@

Submit


Previous
Next
Mark Complete & Next
Powered by HACKTHEBOX