CI passing rules 435 license Apache-2.0

capa detects capabilities in executable files. You run it against a PE file or shellcode and it tells you what it thinks the program can do. For example, it might suggest that the file is a backdoor, is capable of installing services, or relies on HTTP to communicate.

Check out the overview in our first capa blog post.

```
$ capa.exe suspicious.exe

+------------------------+------------------------------------------------
--------------------------------+
| ATT&CK Tactic          | ATT&CK Technique
|
|------------------------+------------------------------------------------
--------------------------------|
| DEFENSE EVASION        | Obfuscated Files or Information [T1027]
|
| DISCOVERY              | Query Registry [T1012]
|
|                        | System Information Discovery [T1082]
|
| EXECUTION              | Command and Scripting Interpreter::Windows
Command Shell [T1059.003]              |
|                        | Shared Modules [T1129]
|
| EXFILTRATION           | Exfiltration Over C2 Channel [T1041]
|
| PERSISTENCE            | Create or Modify System Process::Windows
Service [T1543.003]                      |
+------------------------+------------------------------------------------
--------------------------------+

+-------------------------------------------------------+-----------------
--------------------------------+
| CAPABILITY                                            | NAMESPACE
|
|-------------------------------------------------------+-----------------
--------------------------------|
| check for OutputDebugString error                     | anti-
```

```
analysis/anti-debugging/debugger-detection |
| read and send data from client to server             | c2/file-transfer
|
| execute shell command and capture output             | c2/shell
|
| receive data (2 matches)                             | communication
|
| send data (6 matches)                                | communication
|
| connect to HTTP server (3 matches)                   |
communication/http/client                      |
| send HTTP request (3 matches)                        |
communication/http/client                      |
| create pipe                                          |
communication/named-pipe/create                |
| get socket status (2 matches)                        |
communication/socket                           |
| receive data on socket (2 matches)                   |
communication/socket/receive                   |
| send data on socket (3 matches)                      |
communication/socket/send                      |
| connect TCP socket                                   |
communication/socket/tcp                       |
| encode data using Base64                             | data-
manipulation/encoding/base64              |
| encode data using XOR (6 matches)                    | data-
manipulation/encoding/xor                 |
| run as a service                                     | executable/pe
|
| get common file path (3 matches)                     | host-
interaction/file-system                   |
| read file                                            | host-
interaction/file-system/read              |
| write file (2 matches)                               | host-
interaction/file-system/write             |
| print debug messages (2 matches)                     | host-
interaction/log/debug/write-event         |
| resolve DNS                                          | host-
interaction/network/dns/resolve           |
| get hostname                                         | host-
interaction/os/hostname                   |
| create a process with modified I/O handles and window | host-
interaction/process/create                |
| create process                                       | host-
interaction/process/create                |
| create registry key                                  | host-
interaction/registry/create               |
| create service                                       | host-
interaction/service/create                |
| create thread                                        | host-
interaction/thread/create                 |
| persist via Windows service                          |
persistence/service                            |
+------------------------------------------------------+------------------
--------------------------------+
```

download and usage

Download stable releases of the standalone capa binaries here. You can run the standalone binaries without installation. capa is a command line tool that should be run from the terminal.

To use capa as a library or integrate with another tool, see doc/installation.md for further setup instructions.

For more information about how to use capa, see doc/usage.md.

example

In the above sample output, we ran capa against an unknown binary (`suspicious.exe`), and the tool reported that the program can send HTTP requests, decode data via XOR and Base64, install services, and spawn new processes. Taken together, this makes us think that `suspicious.exe` could be a persistent backdoor. Therefore, our next analysis step might be to run `suspicious.exe` in a sandbox and try to recover the command and control server.

By passing the `-vv` flag (for very verbose), capa reports exactly where it found evidence of these capabilities. This is useful for at least two reasons:

- it helps explain why we should trust the results, and enables us to verify the conclusions, and
- it shows where within the binary an experienced analyst might study with IDA Pro

```
λ capa.exe suspicious.exe -vv
...
execute shell command and capture output
namespace   c2/shell
author      matthew.williams@fireeye.com
scope       function
att&ck      Execution::Command and Scripting Interpreter::Windows Command
Shell [T1059.003]
references  https://docs.microsoft.com/en-
us/windows/win32/api/processthreadsapi/ns-processthreadsapi-startupinfoa
examples    Practical Malware Analysis Lab 14-02.exe_:0x4011C0
function @ 0x10003A13
  and:
    match: create a process with modified I/O handles and window @
0x10003A13
      and:
        or:
          api: kernel32.CreateProcess @ 0x10003D6D
        number: 0x101 @ 0x10003B03
        or:
          number: 0x44 @ 0x10003ADC
        optional:
          api: kernel32.GetStartupInfo @ 0x10003AE4
    match: create pipe @ 0x10003A13
      or:
        api: kernel32.CreatePipe @ 0x10003ACB
    or:
      string: cmd.exe /c  @ 0x10003AED
...
```

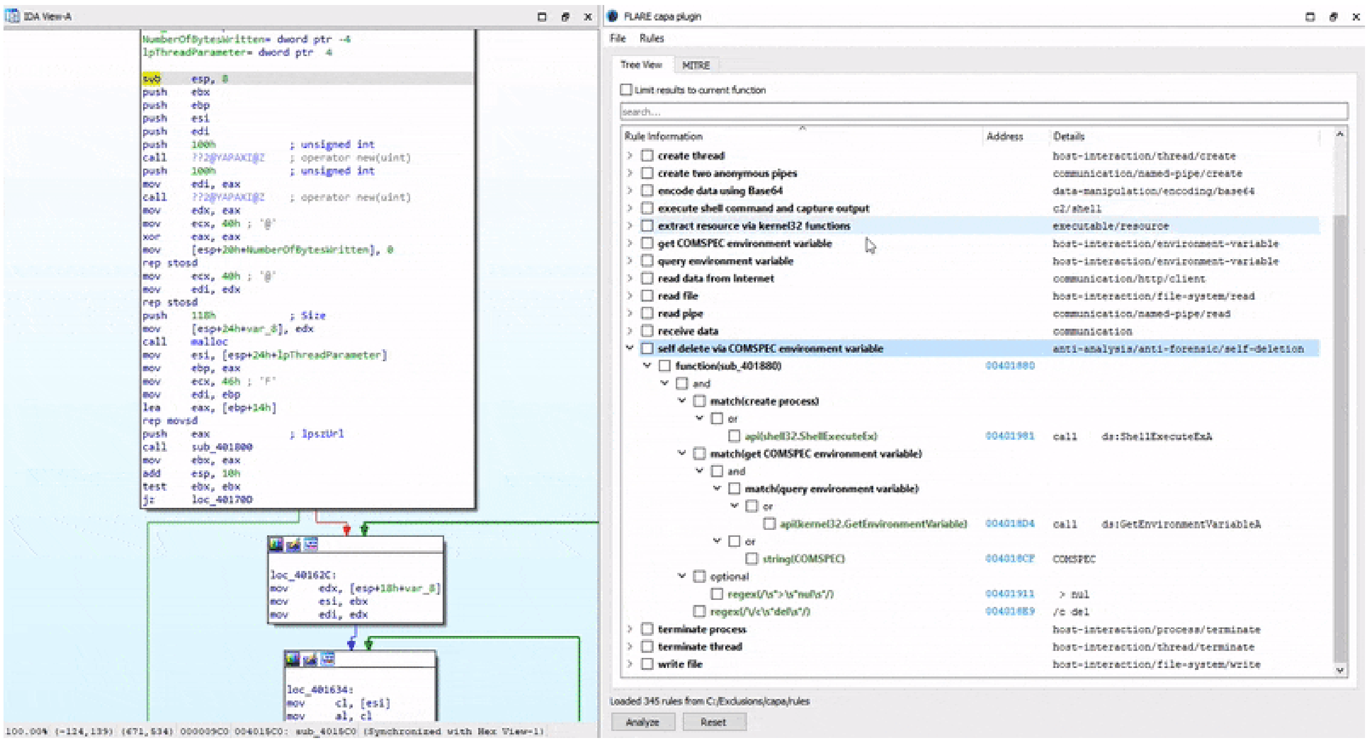
capa uses a collection of rules to identify capabilities within a program. These rules are easy to write, even for those new to reverse engineering. By authoring rules, you can extend the capabilities that capa recognizes. In some regards, capa rules are a mixture of the OpenIOC, Yara, and YAML formats.

Here's an example rule used by capa:

```
rule:
  meta:
    name: hash data with CRC32
    namespace: data-manipulation/checksum/crc32
    author: moritz.raabe@fireeye.com
    scope: function
    examples:
      - 2D3EDC218A90F03089CC01715A9F047F:0x403CBD
      - 7D28CB106CB54876B2A5C111724A07CD:0x402350  # RtlComputeCrc32
  features:
    - or:
      - and:
        - mnemonic: shr
        - number: 0xEDB88320
        - number: 8
        - characteristic: nzxor
      - api: RtlComputeCrc32
```

The github.com/fireeye/capa-rules repository contains hundreds of standard library rules that are distributed with capa. Please learn to write rules and contribute new entries as you find interesting techniques in malware.

If you use IDA Pro, then you use can use the capa explorer IDA plugin. capa explorer lets you quickly identify and navigate to interesting areas of a program and dissect capa rule matches at the assembly level.



```
NumberOfBytesWritten= dword ptr -4
lpThreadParameter= dword ptr  4

sub     esp, 8
push    ebx
push    ebp
push    esi
push    edi
push    100h            ; unsigned int
call    ??2@YAPAXI@Z    ; operator new(uint)
push    100h            ; unsigned int
mov     edi, eax
call    ??2@YAPAXI@Z    ; operator new(uint)
mov     edx, eax
mov     ecx, 40h ; '@'
xor     eax, eax
mov     [esp+20h+NumberOfBytesWritten], 0
rep stosd
mov     ecx, 40h ; '@'
mov     edi, edx
rep stosd
push    118h            ; Size
mov     [esp+24h+var_8], edx
call    malloc
mov     esi, [esp+24h+lpThreadParameter]
mov     ebp, eax
mov     ecx, 46h ; 'F'
mov     edi, ebp
lea     eax, [ebp+14h]
rep movsd
push    eax             ; lpszUrl
call    sub_401800
mov     ebx, eax
add     esp, 10h
test    ebx, ebx
jz      loc_401700

loc_40162C:
mov     edx, [esp+18h+var_8]
mov     esi, ebx
mov     edi, edx

loc_401634:
mov     cl, [esi]
mov     al, cl

100.00% (-124,139) (471,534) 000009C0 004015C0: sub_4015C0 (Synchronized with Hex View-1)
```

```
Rule Information                                   Address    Details
> create thread                                               host-interaction/thread/create
> create two anonymous pipes                                  communication/named-pipe/create
> encode data using Base64                                    data-manipulation/encoding/base64
> execute shell command and capture output                    c2/shell
> extract resource via kernel32 functions                     executable/resource
> get COMSPEC environment variable                            host-interaction/environment-variable
> query environment variable                                  host-interaction/environment-variable
> read data from Internet                                     communication/http/client
> read file                                                   host-interaction/file-system/read
> read pipe                                                   communication/named-pipe/read
> receive data                                                communication
v self delete via COMSPEC environment variable                anti-analysis/anti-forensic/self-deletion
  v function(sub_401880)                           00401880
    v and
      v match(create process)
        v or
            api(shell32.ShellExecuteEx)            00401981   call    ds:ShellExecuteExA
      v match(get COMSPEC environment variable)
        v and
          v match(query environment variable)
            v or
                api(kernel32.GetEnvironmentVariable) 004018D4 call    ds:GetEnvironmentVariableA
          v or
              string(COMSPEC)                      004018CF   COMSPEC
      v optional
          regex(/\s*>\s*nul\s*/)                   00401911    > nul
        regex(/\/c\s*del\s*/)                      004018E9   /c del
> terminate process                                           host-interaction/process/terminate
> terminate thread                                            host-interaction/thread/terminate
> write file                                                  host-interaction/file-system/write

Loaded 345 rules from C:/Exclusions/capa/rules
```

further information

capa

- doc/installation
- doc/usage
- doc/limitations
- Contributing Guide

capa rules

- capa-rules repository
- capa-rules rule format