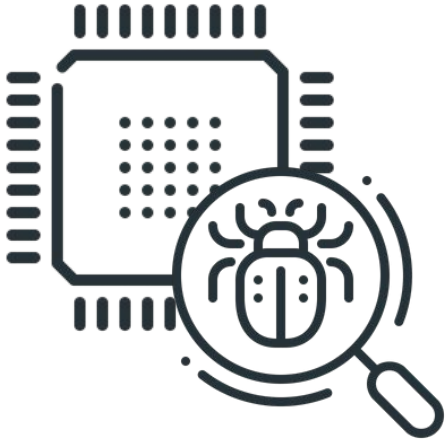
Descriptografando: Experimento em Análise de Memória Volátil Aplicada a Defesa Contra *Ransomware*

Ana Heloísa Bravin Mazur,
Paulo Roberto de Oliveira,
Luciana Andréia Fondazzi Martimiano

Universidade Estadual de Maringá

Visão Geral

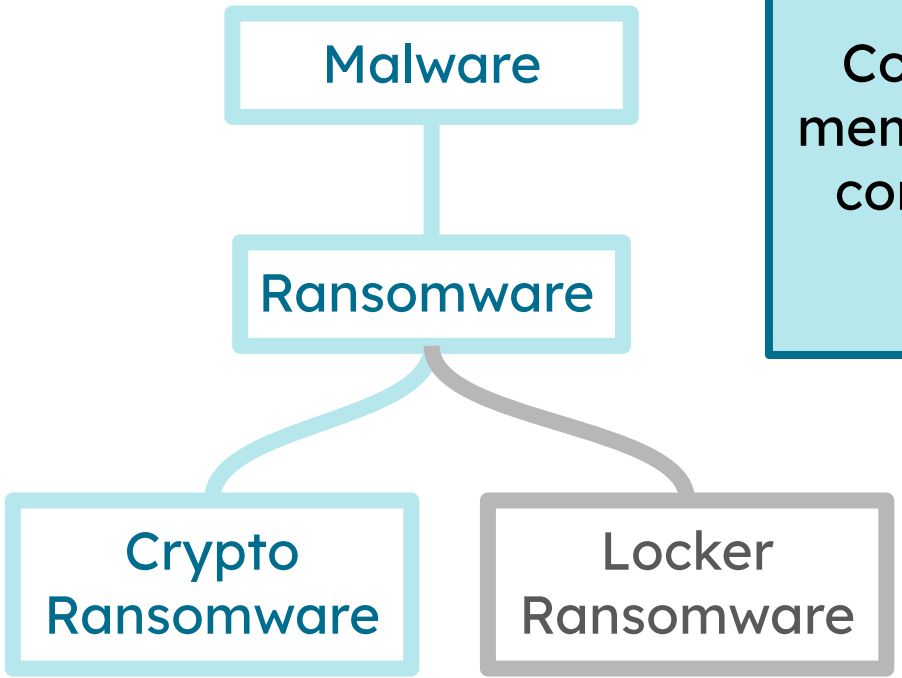
- Introdução
- Metodologia
- Experimento
  - Análise de Memória
  - Análise de Código
  - Identificação de Chaves
- Considerações Finais



Introdução

- Como desenvolver defesas contra *malware*?
  - Analisar ameaças existentes
  - Abordagens de prevenção, detecção e recuperação
- Aquisição e análise da memória RAM de um sistema
  - Recuperação de arquivos e chaves criptográficas em ataques de *ransomware*

Introdução

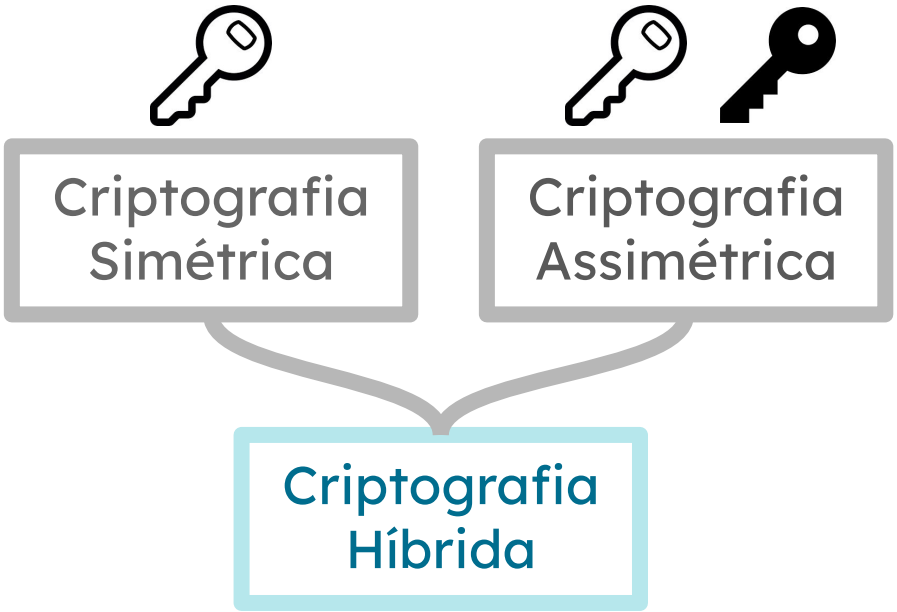


Como a análise de memória pode ser útil contra criptografia maliciosa?

É possível recuperar arquivos atraveś da análise de memória?

Introdução

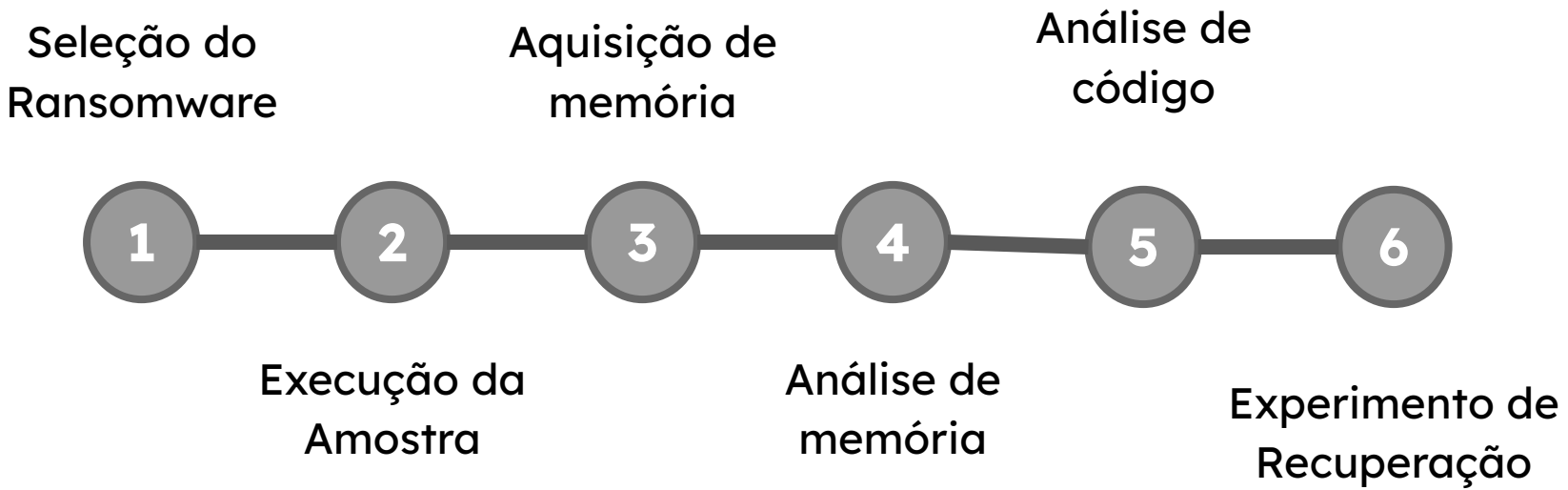
- Identificação de chaves criptograficas
- Permanência da chave na memória
- Gerenciamento de chaves



Identificação de chaves criptograficas

- Busca em força bruta com par *cleartext/ciphertext*
- Relacionamento entre chave pública e chave privada
- Busca por entropia
- Detalhes de implementação da cifra
- Reduzir o espaço de busca através da estrutura da memória

Metodologia



Ferramentas

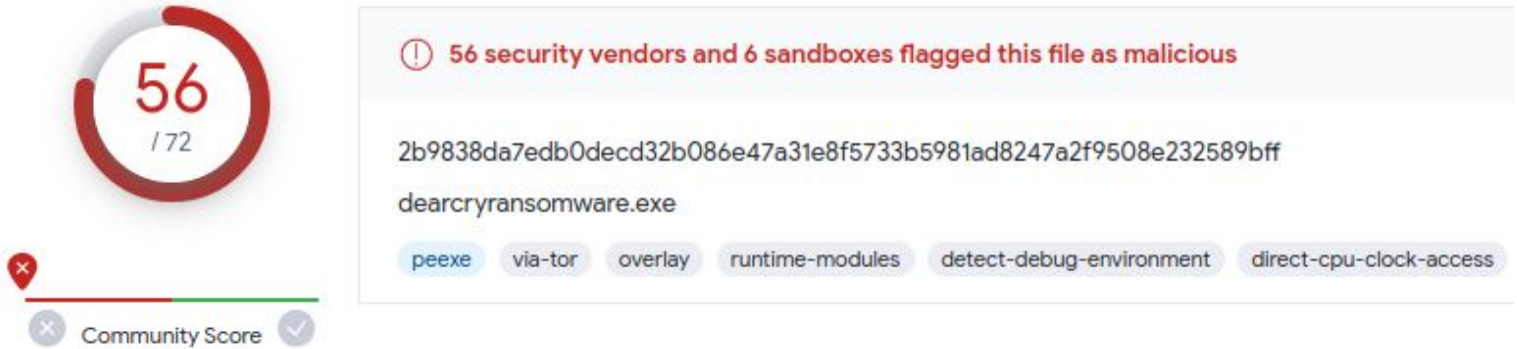
VirtualBox

Volatility 3

Disassembler
Ida

Dearcry – 2021

- Exploit de execução remota de código
- Criptografia híbrida (RSA e AES)



Arquivo sample.txt

```
67 64 6C 6E 64 65 79 67 6A 63 79 68 76 63 61 77 6C 6C 74 70 6E gdlndeygjcyhvcawlltpn
78 6A 6F 6F 69 76 76 6C 76 6D 63 64 76 79 7A 72 68 74 78 76 71 xjooivvlvmcdvyzrhtxvq
78 74 69 75 79 6A 67 78 76 67 6A 69 6E 64 78 6A 77 68 67 6A 66 xtiuyjgxvgjindxjwhgjf
78 67 68 75 64 61 68 6D 72 78 6B 74 67 63 69 76 77 67 67 6E 69 xghudahmrxktgcivwggni
6B 71 71 68 75 61 75 62 73 75 62 61 75 61 76 61 6E 79 74 6F 64 kqqhuaubsubauavanytod
67 75 6B 65 6E 79 64 66 71 73 7A 70 72 69 73 65 6C 6E 76 76 79 gukenydfqszpriselnvvy
6C 68 63 67 7A 78 6D 76 6D 7A 75 62 69 66 75 72 6D 6D 63 75 6E lhcgzxmvmzubifurmmcun
73 71 71 71 6A 78 78 75 76 61 67 70 74 74 6C 6D 6A 78 7A 78 72 sqqqjxxuvagpttlmjxzxr
```

Arquivo sample.txt.CRYPT

```
44 45 41 52 43 52 59 21 00 01 00 00 00 79 17 D9 92 CE 2C 34 49 DEARCRY!.....y....,4I
C3 68 20 B6 4C 8A 29 1C 9A 61 B4 30 88 9A 18 7D 3A 87 7B 11 C9 .h .L.)..a.0...}:.{..
AC C9 81 36 AE D0 E2 CB F1 1C DC 92 FD 14 68 ED A7 F2 65 74 8B ...6..........h...et.
50 F6 3F 60 79 99 18 19 EB 83 8B 34 92 04 99 84 37 1D 7B DE 6C P.?`y......4....7.{.l
8C 5E DA 37 12 69 D7 5E D8 63 C6 DE 4C F2 83 6C 74 18 60 15 7E .^.7.i.^.c..L..lt.`.~
67 93 32 30 94 AF 31 C3 65 FD 1E 35 79 C2 78 B8 45 EB 5F 16 A3 g.20..1.e..5y.x.E._..
72 40 4E B1 33 BB 71 20 69 9D A9 C9 79 D4 4C 91 72 96 DA AC 6B r@N.3.q i...y.L.r...k
2D 00 C8 1C 8D 28 69 BE 43 4F 58 A4 D4 73 70 33 99 F1 42 2F FD -....(i.COX..sp3..B/.
```

Experimento

- Aquisição de memória
- Análise de memória
  - Processos em execução
    - Identificação do processo malicioso
  - Extração de arquivos

PID	PPID	Nome	Endereço Virtual	Threads	Início
4	0	System	0xa6017687f040	106	2023-11-11 20:44:32
72	4	Registry	0xa601769ac040	4	2023-11-11 20:44:17
780	480	cmd.exe	0xa6017e2ce080	1	2023-11-29 22:29:45
5784	708	dllhost.exe	0xa6017d5d3300	7	2023-12-03 17:15:59
5608	1264	audiodg.exe	0xa6017e1ca080	3	2023-12-03 17:17:37
5928	780	0e55ead3b8fd30...	0xa6017d59e300	1	2023-12-03 17:18:34
5404	1216	SearchFilterHo...	0xa6017e5020c0	6	2023-12-03 17:18:42

Tabela 1 – Saída do plugin windows.pslist

Experimento

- Análise de Código
  - Análise Estática Básica
    - Strings
    - DLLs
    - Chave pública
    - Mensagem de resgate

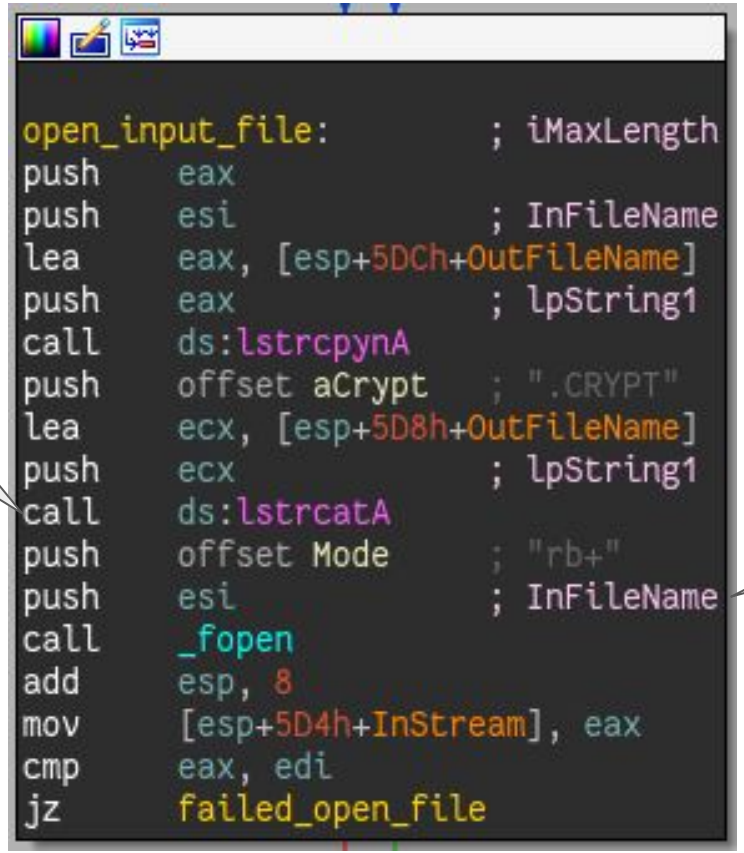
Conteúdo da *string*
OpenSSL
.CRYPT
——BEGIN RSA PUBLIC KEY——
——END RSA PUBLIC KEY——
... .EXE .DLL .CAD .AVI .H.CSV .I

DLL
KERNEL32.dll
ADVAPI32.dll
WS2_32.dll
USER32.dll
CRYPT32.dll

Experimento

- Análise Estática Avançada
  - Assinatura FLIRT OpenSSL
    - Identificação da rotina de criptografia de arquivos
    - Identificação do tipo de criptografia
  - Local da chave na memória
  - Formato do cabeçalho
  - Cópia e destruição do arquivo original

Concatena o nome do arquivo original com a extensão “.CRYPT”



```
open_input_file:          ; iMaxLength
push    eax
push    esi               ; InFileName
lea     eax, [esp+5DCh+OutFileName]
push    eax               ; lpString1
call    ds:lstrcpynA
push    offset aCrypt     ; ".CRYPT"
lea     ecx, [esp+5D8h+OutFileName]
push    ecx               ; lpString1
call    ds:lstrcatA
push    offset Mode       ; "rb+"
push    esi               ; InFileName
call    _fopen
add     esp, 8
mov     [esp+5D4h+InStream], eax
cmp     eax, edi
jz      failed_open_file
```

Abre o arquivo original

```
push    48                  ; Size
lea     eax, [esp+5D8h+RandomBuffer+1]
push    edi                 ; 0
push    eax                 ; void *
mov     byte ptr [esp+5E0h+RandomBuffer], 0
call    _memset
lea     ecx, [esp+5E0h+RandomBuffer]
push    48
push    ecx
call    _RAND_bytes
```

Alocação da chave e vetor de inicialização na pilha, preenchidos com *bytes* aleatórios

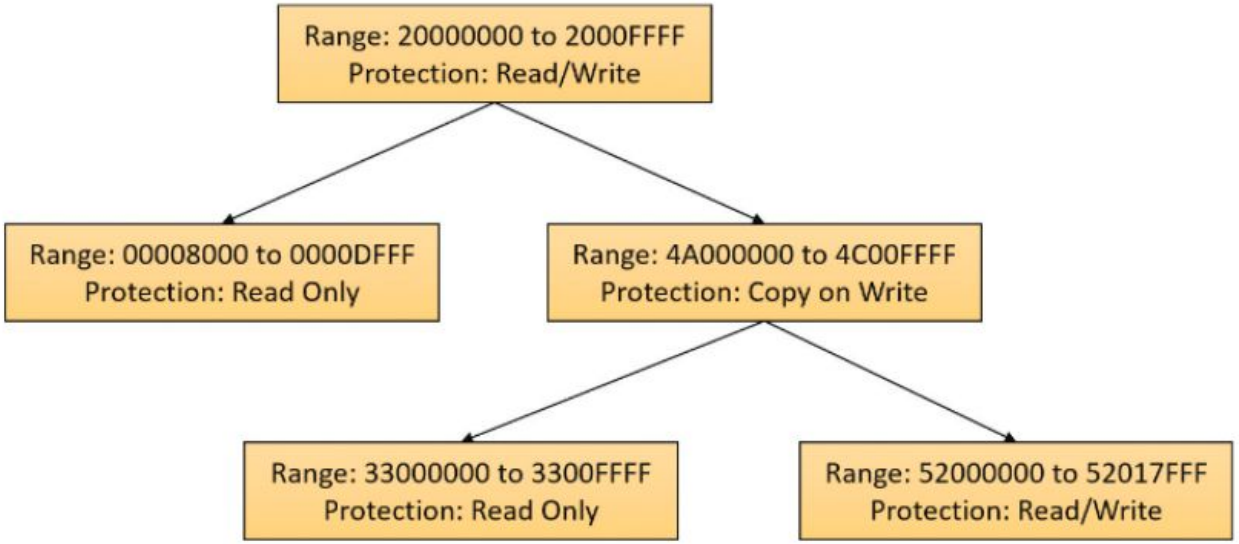
Gravação do início do cabeçalho (marcador “DEARCRY!”)

```
push    esi                 ; OutStream
push    8                   ; ElementCount
push    1                   ; ElementSize
push    offset aDearcry     ; Buffer
call    _fwrite
```

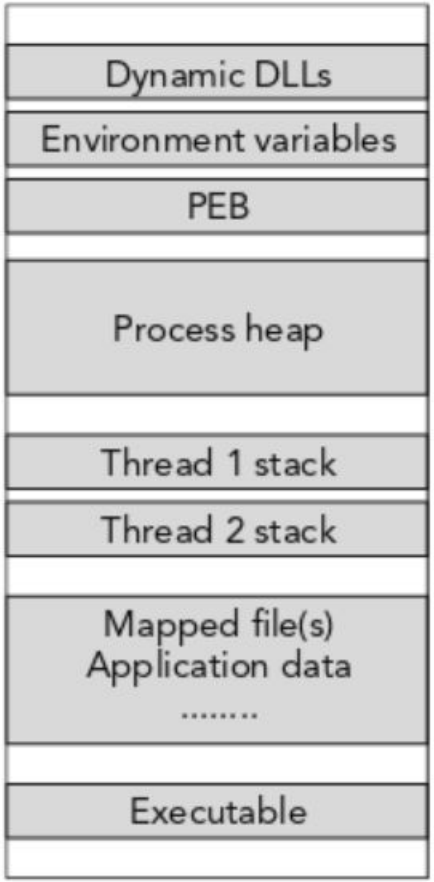
```
44 45 41 52 43 52 59 21 00 01 00 00 69 47 74 F3 20 C3 8A 29 C0
F9 5D DD 2A F4 C4 CD C7 D5 10 0D 3B 92 0F 2F 9C 81 75 FB 1C B0
0D 4D B0 60 2E DE 83 CD 2B 7C 81 00 C0 D3 5D 52 EB 2C 37 E3 2E
06 95 AB 3A 63 ED 16 DC 73 4F EA 51 31 74 89 93 B3 61 69 CA 3C
67 83 AC 55 36 F9 A5 85 47 6B 6A FD 0A C4 CC 19 6F CC 98 ED 7C
B1 FA C0 56 32 56 12 89 4E 68 6B C3 3B 39 22 8E 59 7F 8B C9 EA
81 3A 03 ED 24 D5 D1 43 BB 7B 5B AE 49 58 2F 38 EA EB 3F 73 7A
79 B2 C5 BD 13 2C FA 1B 21 32 61 8B 88 6C C0 3D D9 C3 5B F9 F9
30 AF 9C BE 96 51 34 65 28 04 14 D4 A0 09 55 3F 7D 2B 75 3D 0A
8F 8E 46 2F F6 72 D7 CB F5 D1 5C EE AD C1 39 61 2C BE 30 5E F7
6F 6F 05 F2 34 F7 CC 80 42 E0 0D D3 C9 17 79 B9 1F 85 90 81 B5
53 7F 0E FA CD 29 AF 86 9E 68 D1 37 E7 8B 0C 36 19 35 11 C9 D5
49 B3 7C 93 2F 63 DA FE 81 31 2C 5B F6 0D 45 9C 04 00 00 00 00
04 00 00 00 00 00 00 A2 A1 62 7D 66 CB D4 59 92 5E FB 66 E4 0D
2D 73 24 5D B4 7C 89 3C 43 1E 4A B5 98 AB 91 AC D2 B5 CE DF 63
03 09 BF 1B E0 D8 2C 88 54 3B 2B 95 19 8C 18 42 DF A4 3C FE C9
60 D9 5B FD 4F 19 E0 67 A5 32 56 A8 5B 0D E0 D3 C2 B6 53 4D 4A
```

Experimento

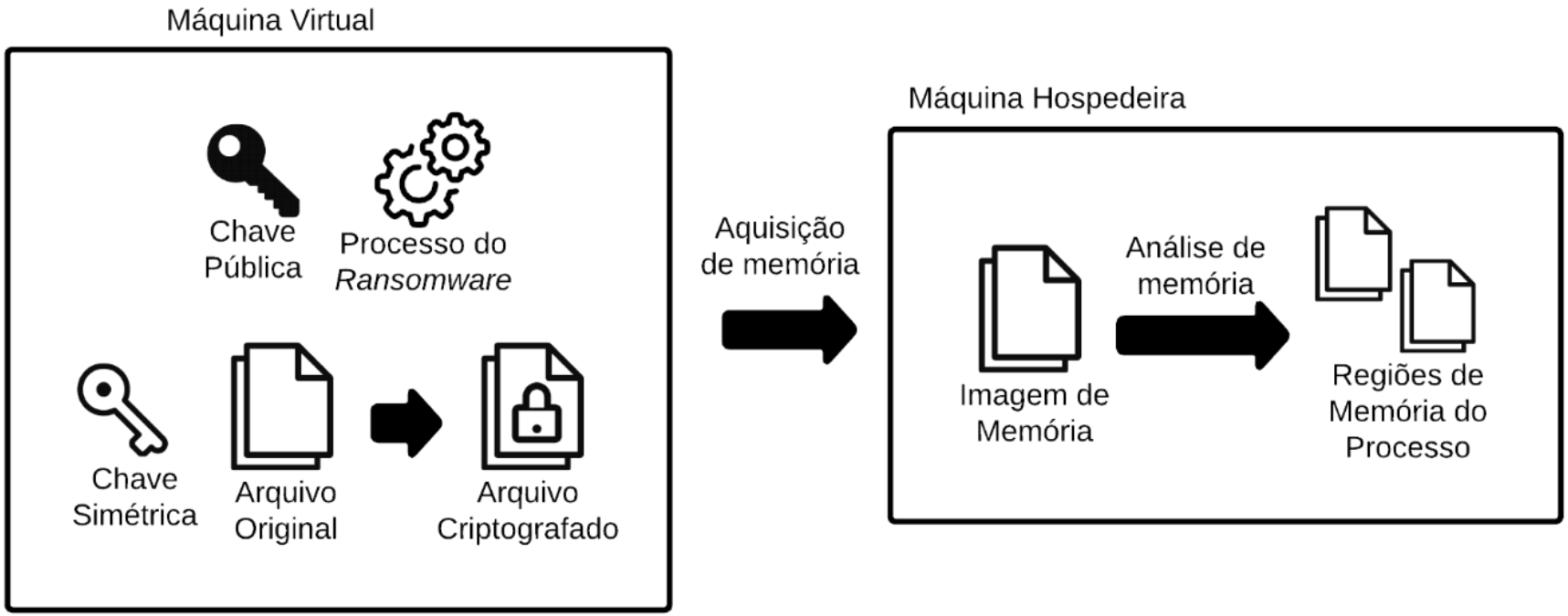
- Identificação de chaves
  - Busca por entropia
    - Estimativa
    - Teste de tipo de arquivo
  - Regiões de memória de pilha
- Recuperação do arquivo

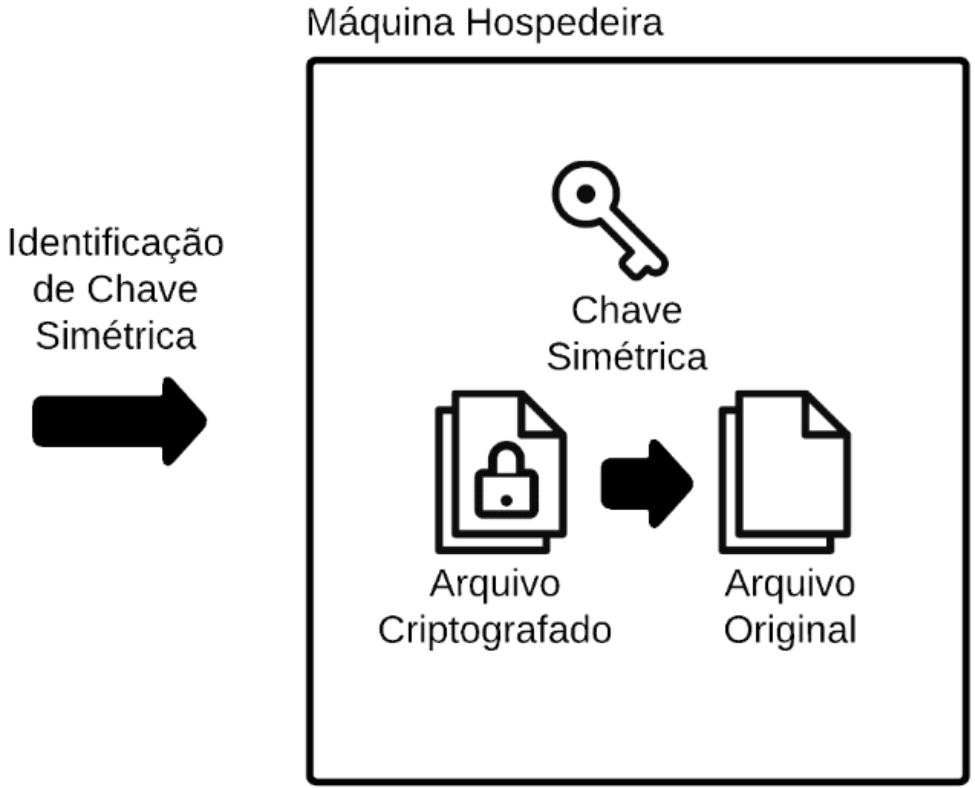


(YOSIFOVICH et al., 2017)



(LIGH et al., 2014)


Máquina Virtual
Chave
Pública
Processo do
Ransomware
Chave
Simétrica
Arquivo
Original
Arquivo
Criptografado
Aquisição
de memória
Máquina Hospedeira
Imagem de
Memória
Análise de
memória
Regiões de
Memória do
Processo


Máquina Hospedeira
Identificação de Chave Simétrica
Chave Simétrica
Arquivo Criptografado
Arquivo Original

Considerações finais

- Demonstração de técnicas de identificação de material criptográfico na memória
- Técnicas de redução do espaço de busca com base na estrutura da memória e análise de código
- Recuperação de apenas um arquivo com criptografia híbrida
    - Memória deve ser adquirida no momento certo

Trabalhos futuros

- Análise e comparação de técnicas de criptografia e gerenciamento de chaves utilizadas por ransomware
- Automatização da identificação de chaves com base na estrutura da memória

Obrigada!

ra118003@uem.br
proliveira2@uem.br
lafmartimiano@uem.br