MALWARES EVASIVOS: TÉCNICAS E DETECÇÃO

5º Fórum Brasileiro de CSIRTS

FELIPE ALMEIDA
FELIPE.ALMEIDA@AXUR.COM

LUCAS MOURA
LUCAS.MOURA@AXUR.COM

Quem somos

- Felipe Almeida

- Lucas Moura

Agenda

1. **Introdução**
2. Definições
3. Ambiente de análise | Cuckoo Sandbox
4. Técnicas de evasão e anti-evasão
5. Fraquezas
6. Estatísticas
7. Conclusões

Contexto

- Análise de malware
- Foco em análise dinâmica
- Técnicas de evasão e anti-evasão

Objetivos

- O que é um malware evasivo?
- Como funciona a análise dinâmica de malware?
- Exemplo de ambiente para análise dinâmica
- Como funcionam as técnicas para evasão de sandboxes?
  - Como detectar e defender-se delas?

Agenda

1. Introdução
2. **Definições**
3. Ambiente de análise | Cuckoo Sandbox
4. Técnicas de evasão e anti-evasão
5. Fraquezas
6. Estatísticas
7. Conclusões

Malware - Tipos de Análise

- Estática
  - *strings*
  - Decompile
    - JVM, .NET CLR
  - Assembly
  - ↑ VBScript (*.vbs, .vbe*), JScript, etc.
- Dinâmica
  - Análise de comportamento
    - Depois da execução, o que o malware faz?
    - Arquivos abertos, atividade de rede, ...?
- Vantagens e desvantagens?
- Automatizar!

Tipos de Malware

- Trojan
- Vírus
- Worm
- Banker
- Dropper
- Evasive
- **Não faremos distinção**

Evasão

- Definição: ato ou processo de evadir; fuga, escapada; evitar algo que não deseja lidar.
- Malware
  - Comportamento sensível ao ambiente
- Primeiras técnicas
  - Anti-debug
- Hoje?
  - Anti-VM
  - [Anti-]anti-vm = **Anti^2-VM**
  - *n* formas [500+; Krugel, 2015]

- Não confundir com VM escape

Agenda

1. Introdução
2. Definições
3. **Ambiente de análise | Cuckoo Sandbox**
4. Técnicas de evasão e anti-evasão
5. Fraquezas
6. Estatísticas
7. Conclusões

Ambiente

- Oracle VirtualBox
  - http://www.virtualbox.org

- Cuckoo Sandbox [modified]
  - http://www.cuckoosandbox.org
  - https://github.com/spender-sandbox/cuckoo-modified

- Host → Fedora 23

- Guest → **Windows 7** [64-bit]

Cuckoo Sandbox

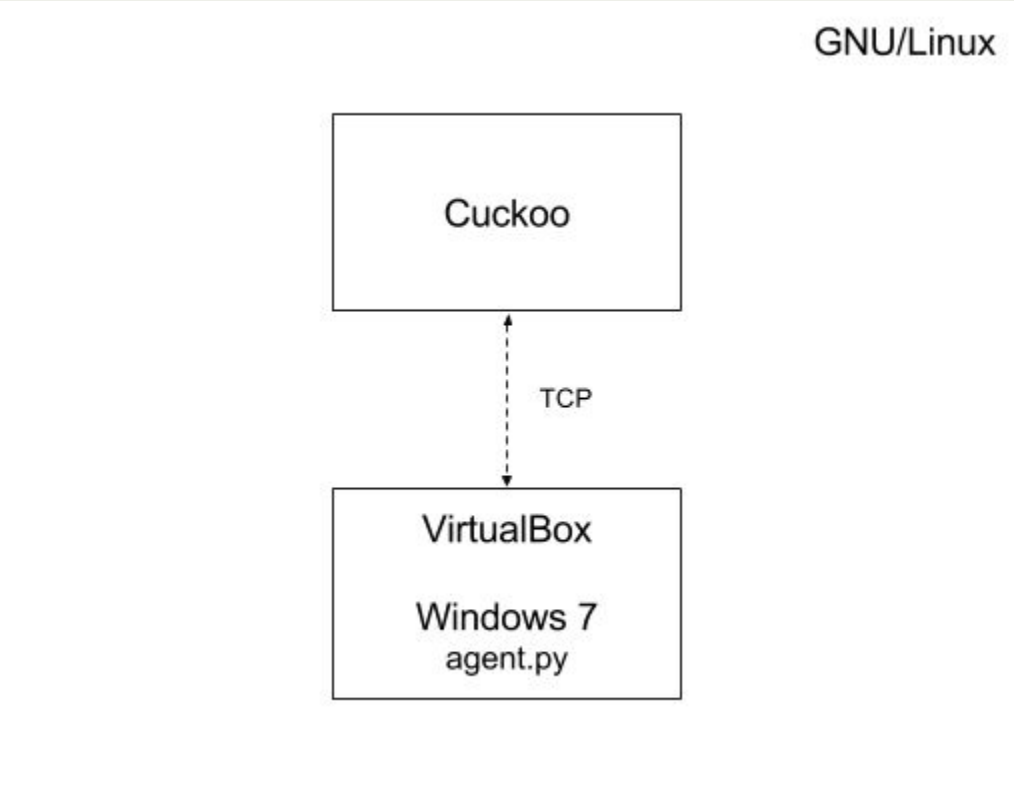
- Análise dinâmica de malwares
- VirtualBox, VMWare, Xen, KVM, etc.
- Roda na VM também
  - cuckoomon.dll
  - agent.py

Cuckoo Sandbox - Relatório

- Chamadas API do sistema
- Arquivos
  - Criados, lidos, modificados
  - Dump
- Serviços
  - Criados, iniciados
- Registro
- Comandos executados
- Tráfego de rede
- Screenshots
- *strings*, Virustotal, volatility, etc.

Cuckoo Sandbox - Análise Exemplo

Cuckoo Sandbox - Comunicação Guest/Host



Cuckoo [simplificado]. Comunicação com o host.

Malwares Evasivos

- O que faz esta string dentro de um malware?

Agenda

1. Introdução
2. Definições
3. Ambiente de análise | Cuckoo Sandbox
4. **Técnicas de evasão e anti-evasão**
5. Fraquezas
6. Estatísticas
7. Conclusões

Técnicas de Evasão (Anti-VM)

E Anti^2-VM também

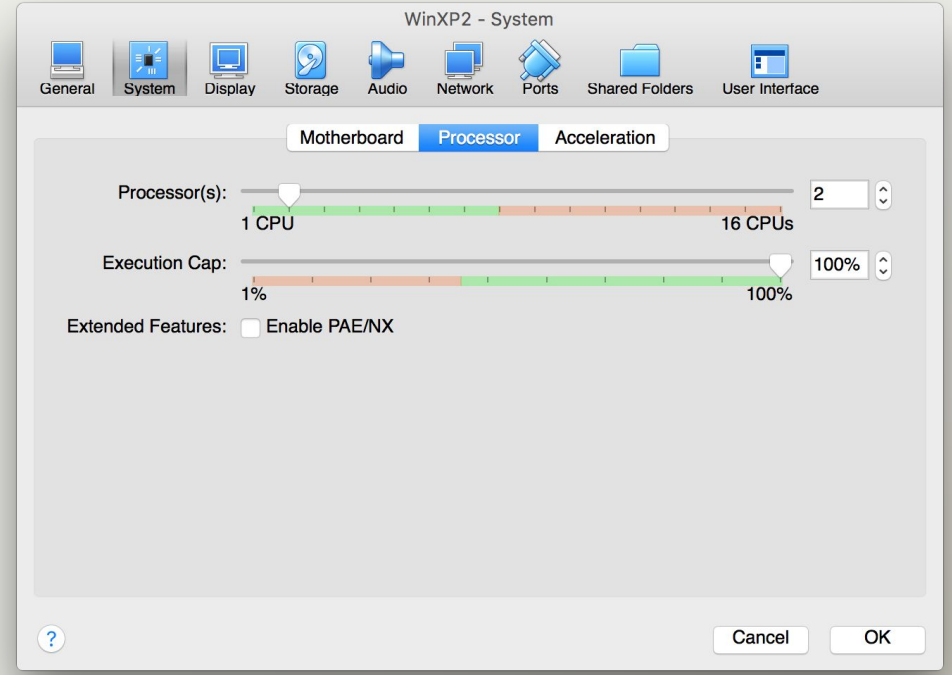
1 - Nro. de processadores

- Utilizada pelo Dyre Wolf em maio de 2015
  - http://www.seculert.com/blogs/new-dyre-version-yet-another-malware-evading-sandboxes

```
int main() {
    SYSTEM_INFO sysinfo;
    GetSystemInfo(&sysinfo);

    int nro_CPU = sysinfo.dwNumberOfProcessors;
    if (nro_CPU == 1) {
      acabou_fui();
      return 0;
     }
    roda_malware();
    return 0;
}
```

[Anti] 1 - Nro. de processadores

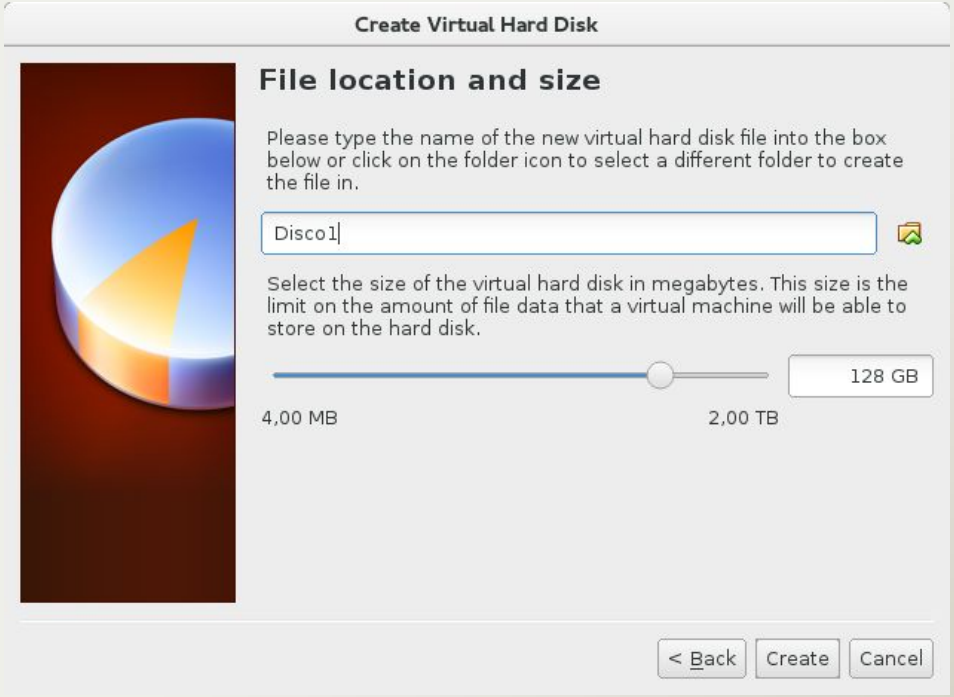


2 - Tamanho do Disco

- Disco < 30GB? VM!

```
int main() {
    ULARGE_INTEGER dskSize;
    unsigned int dskSizeGB = 0;
    if (GetDiskFreeSpaceEx(NULL, NULL, &dskSize, NULL))
        dskSizeGB = dskSize.QuadPart / (1024*1024*1024);
    if (dskSizeGB < 30)
        std::cout << "Estou em uma VM. Fui." << std::endl;
    else
        std::cout << "Executar malware!" << std::endl;
    return 0;
}
```

[Anti] 2 - Tamanho do Disco



3 - Registro (Guest Additions)

- VirtualBox Guest Additions instalada?

```
int main() {
	HKEY hKey;
	LONG res = RegOpenKeyExA(HKEY_LOCAL_MACHINE,
		"SOFTWARE\\Oracle\\VirtualBox Guest Additions", 0, KEY_READ, &hKey);

	if (res == ERROR_SUCCESS)
		std::cout << "Encontrei a chave. Fui." << std::endl;
	else
		std::cout << "Rodando Malware." << std::endl;
}
```

[Anti] 3 - Registro (Guest Additions)

- Solução? Não instalar Guest Additions :)

- Quero/preciso
  - Solução da técnica 4

4 - Outras chaves no Registro

- Outras chaves no registro estão presentes
  - Mesmo sem a instalação do Guest Additions

- (HKEY_LOCAL_MACHINE, "HARDWARE\\Description\\System", **"VideoBiosVersion", "VIRTUALBOX"**)
- (HKEY_LOCAL_MACHINE, "HARDWARE\\ACPI\\DSDT**VBOX**__")
- (HKEY_LOCAL_MACHINE, "HARDWARE\\DESCRIPTION\\System", "SystemBiosDate", "**06/23/99**")
- (HKEY_LOCAL_MACHINE, "HARDWARE\\DEVICEMAP\\Scsi\\Scsi Port 0\\Scsi Bus 0\\Target Id 0\\Logical Unit Id 0", "Identifier", "**VBOX**")
- (HKEY_LOCAL_MACHINE, "SYSTEM\\ControlSet001\\Services**VBoxGuest**")
- (HKEY_LOCAL_MACHINE, "SYSTEM\\ControlSet001\\Services**VBox***")

[Anti] 4 - Outras chaves no Registro

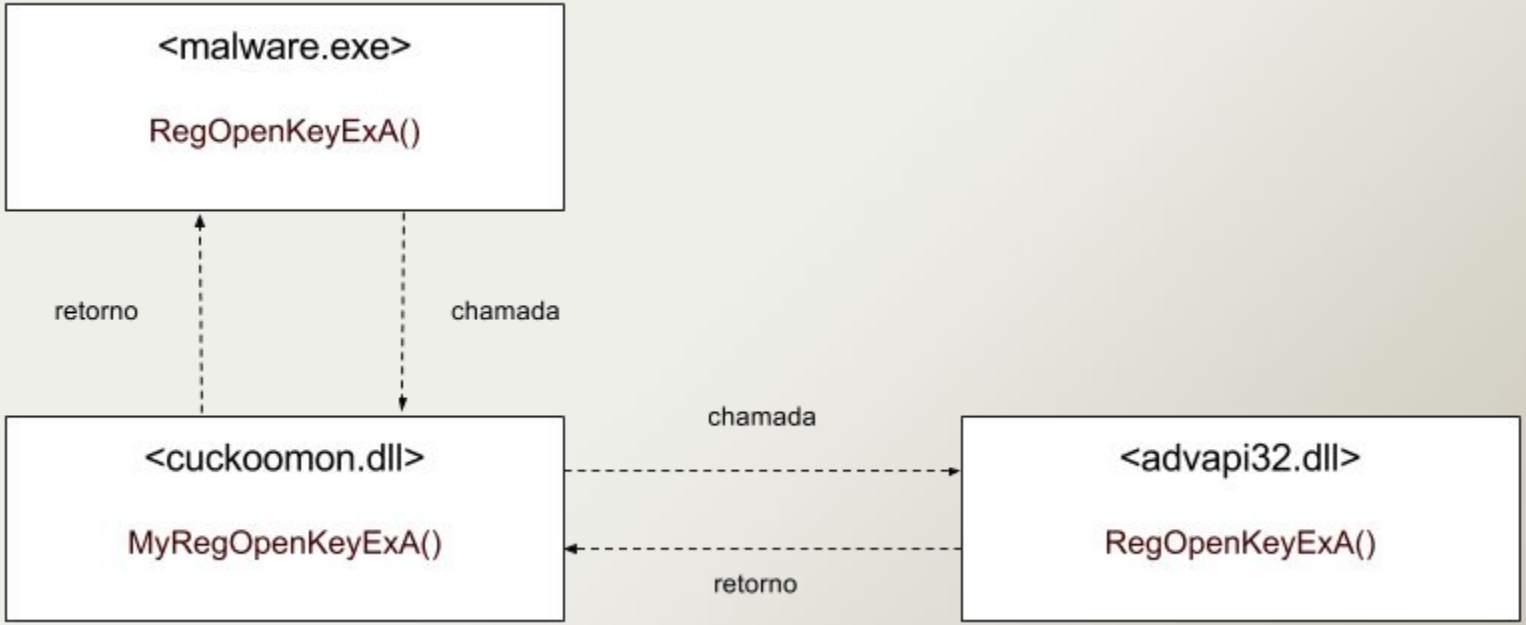
- Solução? **Hook!**

- Interceptar chamadas para API do Windows

- Modificar a resposta de acordo com os interesses [Anti-VM]
  - E logar essas chamadas

- Cuckoo injeta DLL no processo
  - cuckoomon.dll

[Anti] 4 - Outras chaves no Registro



5 - MAC Address

- 3 primeiros bytes do MAC Address → OUI
  - Organizationally unique identifier
  - Identifica o fabricante
  - http://standards-oui.ieee.org/oui.txt

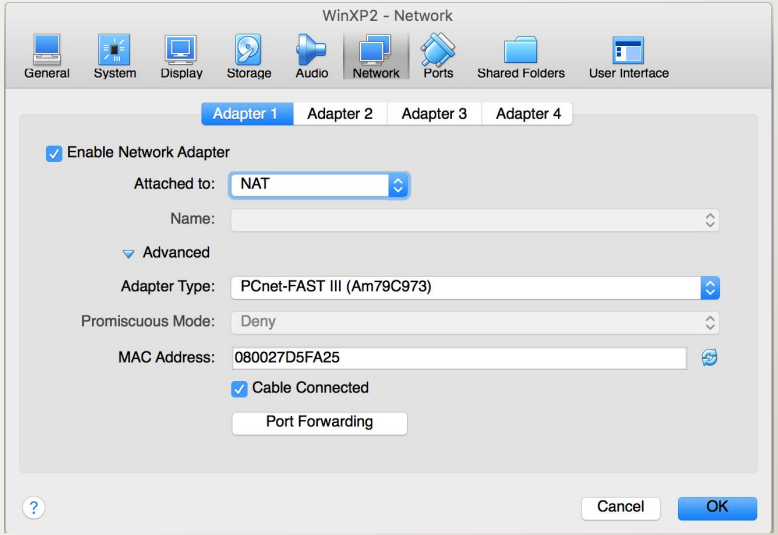
- VirtualBox → [08:00:27]
  - Cadmus Computer Systems
- VMWare → [00:05:69, 00:0C:29, 00:1C:14, 00:50:56]
  - VMware, Inc.

[Anti] 5 - MAC Address

- Troque o MAC da(s) interface(s) de rede da VM

```
$ perl -e 'for ($i=0;$i<6;$i++){@m[$i]=int(rand(256));} printf "%X:%X:%X:%X:%X:%X\n",@m;'
```

- http://www.miniwebtool.com/mac-address-generator/



6 - Atividade do Mouse

- Há movimento no mouse durante um intervalo de tempo?
  - Não? VM!!
  - Sim? #rodarmalware

```
int gensandbox_mouse_act() { // https://github.com/a0rtega/pafish/blob/master/pafish/gensandbox.c
	POINT position1, position2;
	GetCursorPos(&position1);
	Sleep(2000); /* Sleep time */
	GetCursorPos(&position2);
	if ((position1.x == position2.x) && (position1.y == position2.y)) {
		// Sem atividade durante sleep. Fui.
	}
	else {
		// Atividade durante o sleep. Malware em ação.
	}
}
```

[Anti] 6 - Atividade do Mouse

- analyzer/windows/modules/auxiliary/human.py

```
def move_mouse():
    x = random.randint(0, RESOLUTION["x"])
    y = random.randint(0, RESOLUTION["y"])

    USER32.SetCursorPos(x, y)

class Human(Auxiliary, Thread):
    def run(self):
        ....
        # only move the mouse 50% of the time, as malware can choose to act on an
        # "idle" system just as it can on an "active" system
        if random.randint(0, 3) > 1:
            click_mouse()
            move_mouse()
# https://github.com/brad-accuvant/cuckoo-modified/blob/master/analyzer/windows/modules/auxiliary/human.py
```

7 - Sleep

- Processo chama a função *sleep()* e “dorme” por um tempo

- Normalmente a análise possui um *timeout*
  - Após este limite de tempo, a análise é encerrada

[Anti] 7 - Sleep

- Hooking nas funções de *sleep()*
  - NtDelayExecution, NtSetTimer, NtSetTimerEx, etc.

- O que era para levar “muitos segundos” reduz-se a zero

[Anti] 7 - Sleep

Resultado da análise - malware tentou diversas evasões

8 - O *sleep()* foi alterado?

- Função *GetTickCount()* retorna o número de milisegundos desde que o sistema foi iniciado
- Verificando este valor antes e depois do *sleep()* podemos saber se "pulamos no tempo"

```
int gensandbox_sleep_patched() { // pafish/gensandbox.c
        DWORD time1;

        time1 = GetTickCount();
        Sleep(500);
        if ((GetTickCount() - time1) > 450 )
            e();
        else
            exit(0);
}
```

[Anti] 8 - O *sleep()* foi alterado?

- Hook ***GetTickCount()*** **!**

```
HOOKDEF(DWORD, WINAPI, GetTickCount, void) {
    DWORD ret = Old_GetTickCount();

    // add the time we've skipped
    ret += (DWORD)(time_skipped.QuadPart / 10000);

    return ret;
}

// https://github.com/brad-accuvant/cuckoomon-modified/blob/MSVC/hook_sleep.c
```

9 - Processos

- Verificar a lista de processos do sistema operacional

- Roda *python.exe*? *VboxSVC.exe*? *vboxtray.exe*?

[Anti] 9 - Processos

```
protected_procname_list = [
    "vmwareuser.exe",
    "vmwareservice.exe",
    "vboxservice.exe",
    "vboxtray.exe",
    "sandboxiedcomlaunch.exe",
    "sandboxierpcss.exe",
    "procmon.exe",
    "regmon.exe",
    "filemon.exe",
    "wireshark.exe",
    "netmon.exe",
    "prl_tools_service.exe",
    "prl_tools.exe",
    "prl_cc.exe",
    "sharedintapp.exe",
    "vmtoolsd.exe",
    "vmsrvc.exe",
    "python.exe",
    "perl.exe",
]

HIDE_PIDS = set(self.pids_from_process_name_list(protected_procname_list))
```

analyzer/windows/analyzer.py

Outras técnicas

- Há muito mais!
  - Nome de usuário, nome da máquina, "shared folders", detectar GPU, …
  - Técnicas podem gerar falsos-positivos

- Há *n* formas de utilizar a mesma técnica Anti-VM
  - O mesmo serve para as Anti^2-VM

- Confira: https://github.com/a0rtega/pafish

Análise exemplo - Pafish

- Combina diversas técnicas de evasão

- Boa forma para testar a sandbox

Agenda

1. Introdução
2. Definições
3. Ambiente de análise | Cuckoo Sandbox
4. Técnicas de evasão e anti-evasão
5. **Fraquezas**
6. Estatísticas
7. Conclusões

Fraquezas/Dificuldades

- A própria evasão
  - *n* formas
- Hooking
  - Guerra constante
  - Kernel malware
  - Hardcoded
- Automação
  - Crashes?

Kernel Malware

- Hooking em user mode / malware em kernel mode
  - Funções chamadas não serão interceptadas
- Não tão comum
  - Windows atuais carregam apenas drivers assinados digitalmente
- Driver para monitoramento do kernel
  - Zer0m0n: https://github.com/conix-security/zer0m0n

Agenda

1. Introdução
2. Definições
3. Ambiente de análise | Cuckoo Sandbox
4. Técnicas de evasão e anti-evasão
5. Fraquezas
6. **Estatísticas**
7. Conclusões

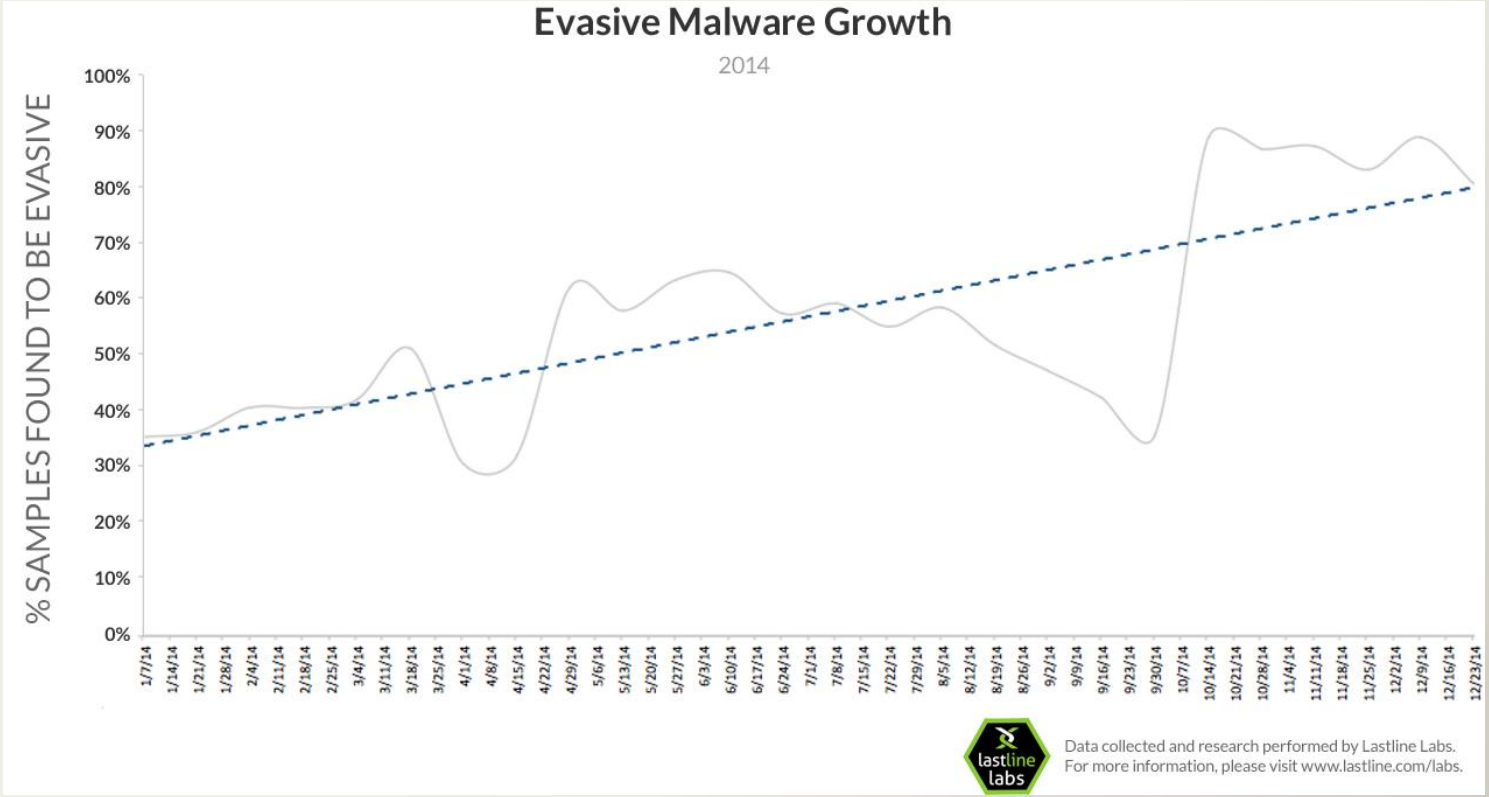
Estatísticas

- Axur
  - Últimos 500 malwares analisados (foco no mercado brasileiro)

- 88% fazem uso de pelo menos uma técnica Anti-VM
  - Mais popular: *sleep()*

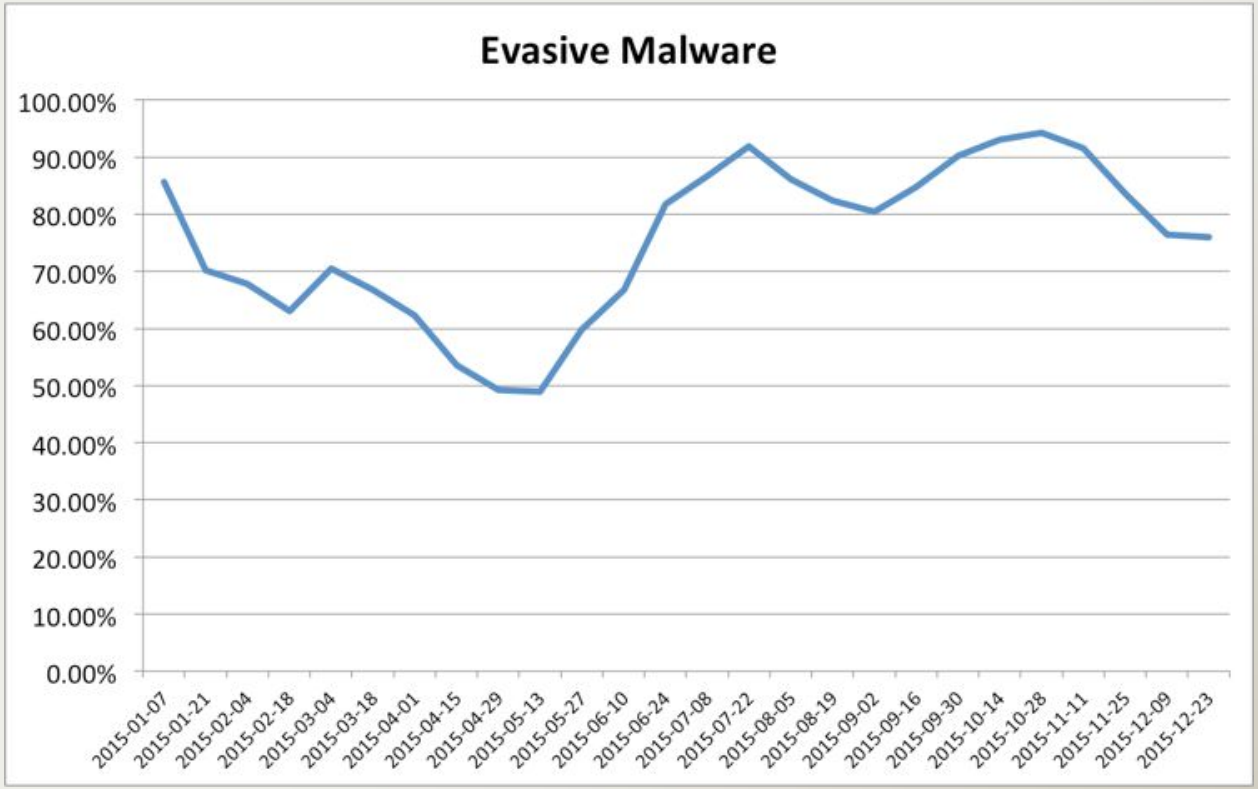
- Destes, 76% utilizam mais de uma técnica

Estatísticas



http://labs.lastline.com/evasive-malware-gone-mainstream

Estatísticas



http://labs.lastline.com/three-interesting-changes-in-malware-activity-over-the-past-year

Agenda

1. Introdução
2. Definições
3. Ambiente de análise | Cuckoo Sandbox
4. Técnicas de evasão e anti-evasão
5. Fraquezas
6. Estatísticas
7. **Conclusões**

Conclusões

- Análise dinâmica? Tem que levar em conta evasão
  - Vale também para análises não-automatizadas em VM

- Futuro
  - Combinações de técnicas evasivas mais elaboradas
  - Criatividade não tem limites :)

Outras Sandboxes

- Malwr
  - https://malwr.com/
- Joe Sandbox
  - https://www.file-analyzer.net/
- Hybrid Analysis
  - https://www.hybrid-analysis.com/

Referências

- Kruegel, Christopher. "Evasive Malware Exposed and Deconstructed." RSA Conference 2015.
  - https://www.rsaconference.com/writable/presentations/file_upload/crwd-t08-evasive-malware-exposed-and-deconstructed.pdf

- Martina Lindorfer, Clemens Kolbitsch, and Paolo Milani Comparetti. 2011. Detecting environment-sensitive malware. In Proceedings of the 14th international conference on Recent Advances in Intrusion Detection (RAID'11), Robin Sommer, Davide Balzarotti, and Gregor Maier (Eds.).
  - http://www.syssec-project.eu/m/page-media/3/disarm-raid11.pdf

- Todos outros sites já citados na apresentação

Muito Obrigado!

Perguntas?!

FELIPE ALMEIDA
FELIPE.ALMEIDA@AXUR.COM

LUCAS MOURA
LUCAS.MOURA@AXUR.COM