

# Automated In-Network Malware Analysis

Why Virtual Machines Can Sputter and Miss

by Gunter Ollmann, VP of Research, Damballa

*Virtualization, emulation and sandbox environments are no match for today's advanced malware.*

## Executive Summary

With the decreasing effectiveness of end point protection suites, information security professionals are turning to alternative approaches to detecting the presence of advanced malware. One approach seeks to analyze inbound suspicious files by allowing them to run in a virtual machine environment – with the hopes of positively identifying malware and extracting forensics data to aid remediation tasks or create signatures to block infection vectors.

The approach is well known to the operators of botnets and cyber crime; and workarounds have been increasingly popular. As such, there are a variety of techniques the bad guys can pull from in order to evade detection by virtualization environments.

This paper is a primer on malware virtualization and the techniques criminals use to detect them and evade analysis. The techniques include:

- Good-Guy Blacklisting - Criminal operators maintain a Good-Guy Blacklist of domains having a history of attempting malware analysis. Knowing these to be antivirus vendors and the like, criminal operators avoid distributing malware to these analysis systems.
- Detecting Blocked Network Access - There are countless techniques used by the criminals to check for legitimate network or Internet access as a means of detecting that the malware is running in an analysis environment. Failing to detect a legitimate freedom to roam, the malware will cease operation in order to hide its malicious behavior, and in fact can actually fake benign behavior to evade analysis - and successfully target legitimate victims.
- Leveraging Unblocked Network Access - The dangers of allowing suspicious binaries to execute in a virtual environment within the enterprise with unfettered access to the Internet or the corporate network should be obvious. And if anyone feels lucky enough to try, the criminals may use alternate means to determine if the asset they are infecting is legit. If found to be an analysis environment criminals may launch retribution attacks.

Bottom line – criminals are very good at assessing the legitimacy of the victim machine they are infecting. Virtualization, emulation and sandbox environments are no match for today's advanced malware. The balance of this paper explores several of the reasons why.

## Introduction

With increasing regularity, modern criminal malware seeks to verify its successful installation on a victim's computer or mobile device before initiating its core malicious functions. Just as there are many well studied and established techniques that malware authors can draw from in determining whether their software agent has successfully penetrated the victim's defenses, there are also several techniques commonly used to detect whether the infected computer is in fact an analysis system used for the automatic analysis of malware.

These later checks are of growing importance to criminal operators. The longer the period in which they can distribute a particular family of malware and infect victims without detection, the more profitable the endeavor becomes for them. By being able to automatically detect whether their malware has been caught by an analysis system and is subsequently acting benign, the malware author can postpone the development of detection signatures capable of thwarting the spread of their malware.

For the last decade, virtualization and emulation technologies have been at the forefront of large-scale automated malware analysis. By instrumenting a dynamic representation (or instance) of a popular operating system and intentionally infecting it with a captured malware sample, it is generally feasible to monitor the activities of the software agent and determine its maliciousness. In some cases it is also possible to automatically create clean-up scripts capable of removing future real-life infections.

In the cat and mouse game of malware development versus detection, malware authors have included their own detection and evasion techniques designed to thwart virtualization and emulation analysis systems. For malware created using popular DIY construction kits, or malware undergoing serial variant production processes and armoring, these detection and evasion techniques have effectively become tick-box options in the production process. Meanwhile, malware researchers and antivirus vendors have been striving to make it more difficult for captured malware samples to identify their instrumented analysis systems.

Since mid-2009, botnet malware developers have been applying additional network-based techniques to determine the "authenticity" of the system targeted for infection. Key amongst these techniques is the requirement to verify Internet connectivity and the presence of any filtering or modification of traffic. This paper investigates some of the more common techniques being employed by criminal botnet operators and what the consequences are for both automated threat analysis and defensive strategies.

## Automated Malware Analysis

In order to understand the current generation of techniques being used to detect and thwart the "good-guy" analysis of malware, it is important to understand some of the automated analysis scenarios that malware authors and criminal botnet operators are trying to evade.

1. Antivirus vendors receive a steady stream of suspicious binary files and run a series of tests to determine whether a file may potentially be malicious. One critical test includes the controlled execution of the suspicious file within a heavily monitored and instrumented instance of an

"average" customer's computer. Based on defined malicious file and system activities, the sample may be labeled as malware and categorized as a specific threat category (e.g. Trojan, backdoor, bot agent, etc.). Study of the malware sample is limited to a short period of time - usually measured in terms of a few seconds or minutes. These environments often do not have access to the Internet.

2. For the analysis of malware that requires Internet access to function, specialized analysis environments filter and modify network traffic in order to trick the malware into believing that it has access in order to identify key characteristics of the malware (e.g. IP or domain name blacklist creation, spam update characteristics, data dropper destinations, etc...). In advanced analysis research labs, the computers being infected with the malware sample may be of the "bare metal" type - which means that for all intents and purposes, the victim computer is real (i.e. not virtualized or emulated) and not instrumented - and therefore contains no local host hints to its analysis role.
3. For large businesses seeking to fill a gap between their desktop antivirus solutions and mail-gateway attachment inspection technologies, the last decade has shown an increasingly popular strategy to employ perimeter malware inspection appliances. These appliances typically perform a series of network heuristic, signature-based checking and virtualized execution processes to determine the maliciousness of the binary file. In most cases, the examination is conducted "out of band" and does not allow Internet access to the file being tested.
4. Honeypot and Honeynet projects focus on infecting a victim computer and performing long-term observations of how the malware is controlled and updated by its criminal masters - typically requiring restricted Internet access. In many cases these infected computers will similarly be "bare metal" systems - and analysis of the malicious activities on the system tends to be more manual than other analysis techniques.

Dealing with malware that requires access to the Internet is troublesome. Without unrestricted access to the Internet the malware sample may not execute correctly or may act benign. If unrestricted Internet access is allowed, malware may automatically commence an attack against some remote target - which may have legal consequences for the analyst.

## Good-Guy Blacklisting

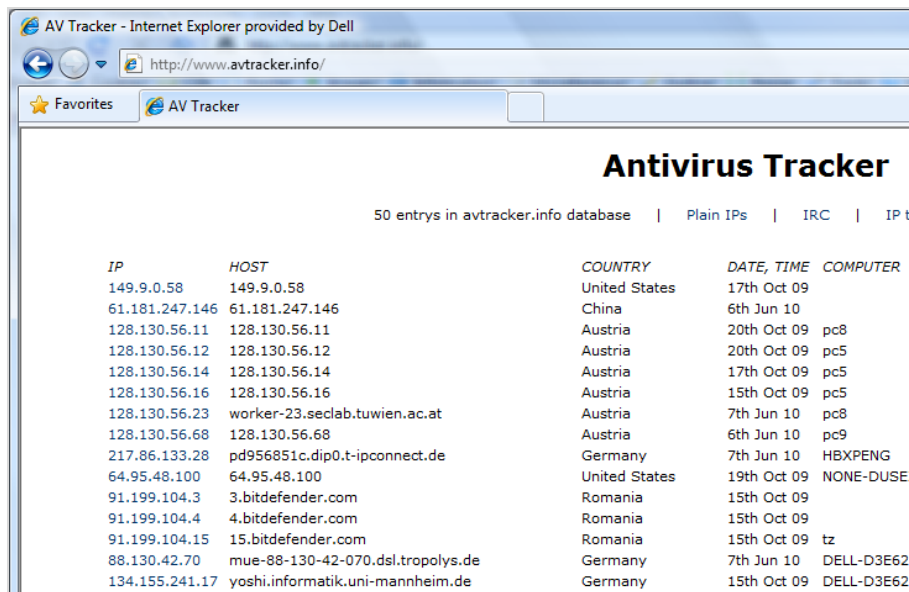
Implementing techniques similar to the blacklisting of IP addresses and domain names used by network protection technologies to defeat certain classes of attack and fraud, criminal operators also maintain their own blacklists of security vendors and research organizations. These "good-guy" blacklists are used in a number of distinct ways:

- **Don't Serve** - Criminals use a wide variety of methods to exploit vulnerable computer systems and serve up malware components that will eventually be installed on the victim's computer. In the case of drive-by-download sites, exploit engines running on the malicious server will check the IP address of the intended victim against a blacklist and decide on whether exploits and malware will actually be served. The purpose of this blacklist approach is to prevent samples of

malware being captured by antivirus vendors and to remain undetected by web scanning engines designed to flag and block malicious websites.

- **Don't Answer** - When malware has been installed on a victim's computer it will often be primed to connect to a remote server controlled by the criminal for the purpose of receiving new malware updates or joining a botnet. The purpose of a blacklist in this scenario is to identify whether malware researchers are analyzing malware samples and trying to locate caches of information or command-and-control services associated with the criminal operator of the malware and, once identified as such, to not answer their network requests. Doing so makes it considerably more difficult for the good-guys to identify whether an attack is "live" and ongoing, or has already been shut down, and prevents any further information leakage about the criminal operator(s).
- **Be Nice** - As an alternative to not serving malicious content or answering requests to the server, some criminal operators choose to adopt a chameleon mode and present false and clearly benign material to requests from good-guy IP addresses contained in their blacklist. The type of alternative content being served can range from faked copies of domain holding pages and cloned blog sites, through to redirection on to popular legitimate sites.

Access to these "good-guy blacklists" is fairly easy. Many hacker and botnet forums provide access to compiled lists of IP addresses and netblocks associated with commercial security vendors - which are updated and shared freely among forum members. In addition, a number of free Websites have sprung up in recent years that provide tracking services and openly publish the lists.



AV Tracker - Internet Explorer provided by Dell  
 http://www.avtracker.info/  
 AV Tracker

### Antivirus Tracker

50 entries in avtracker.info database | Plain IPs | IRC | IP t

| IP             | HOST                              | COUNTRY       | DATE, TIME  | COMPUTER   |
|----------------|-----------------------------------|---------------|-------------|------------|
| 149.9.0.58     | 149.9.0.58                        | United States | 17th Oct 09 |            |
| 61.181.247.146 | 61.181.247.146                    | China         | 6th Jun 10  |            |
| 128.130.56.11  | 128.130.56.11                     | Austria       | 20th Oct 09 | pc8        |
| 128.130.56.12  | 128.130.56.12                     | Austria       | 20th Oct 09 | pc5        |
| 128.130.56.14  | 128.130.56.14                     | Austria       | 17th Oct 09 | pc5        |
| 128.130.56.16  | 128.130.56.16                     | Austria       | 15th Oct 09 | pc5        |
| 128.130.56.23  | worker-23.seclab.tuwien.ac.at     | Austria       | 7th Jun 10  | pc8        |
| 128.130.56.68  | 128.130.56.68                     | Austria       | 6th Jun 10  | pc9        |
| 217.86.133.28  | pd956851c.dip0.t-ipconnect.de     | Germany       | 7th Jun 10  | HBXPENG    |
| 64.95.48.100   | 64.95.48.100                      | United States | 19th Oct 09 | NONE-DUSE  |
| 91.199.104.3   | 3.bitdefender.com                 | Romania       | 15th Oct 09 |            |
| 91.199.104.4   | 4.bitdefender.com                 | Romania       | 15th Oct 09 |            |
| 91.199.104.15  | 15.bitdefender.com                | Romania       | 15th Oct 09 | tz         |
| 88.130.42.70   | mue-88-130-42-070.dsl.tropolys.de | Germany       | 7th Jun 10  | DELL-D3E62 |
| 134.155.241.17 | yoshi.informatik.uni-mannheim.de  | Germany       | 15th Oct 09 | DELL-D3E62 |

Figure: A free "Antivirus Tracker" service listing IP addresses associated with antivirus vendors that automatically analyze malware.

## Blacklist Creation

There are, of course, a wide range of techniques that can be used by criminals to compile good-guy blacklists. Some of the more common methods include:

- **Traffic Analysis** - Automated malware analysis systems are designed to process high volumes of captured malware samples. If the malware author or botnet operator has released multiple variants or families of malware (including version number tracking), they will quite easily see the same computers being compromised over and over again as they "phone home." Simple tracking of the IP address or host names (or even the unique ID of the computer being infected) will quickly yield a list of vendor-related malware analysis systems.
- **Handshake Anomalies** - In the process of investigating suspicious domains and IP addresses associated with malware and other malicious activities, multiple network-based investigation techniques may be used by security vendors. Investigative activities include connecting to the criminal's server to check what type of Internet services it has running and querying domain registration details associated with the administration of the server. These activities can be easily detected - particularly those attempts to manually (or generically) connect to services associated with the malware without providing the correct credentials or appropriate handshake verification.
- **Seeding** - Practically all antivirus vendors share or consume malware samples amongst each other. Many of these samples come via free malware testing portals. Malware authors may submit a unique malware sample to one of the free testing portals with the expectation that antivirus vendors will eventually receive a copy and will attempt to automatically analyze it. By submitting a malware sample that has never been distributed "in the wild," any subsequent attempts to connect to or investigate the malware's phone-home server could only have come from antivirus vendors and security researchers - thereby enumerating the systems owned and operated by the good guys.

## Blocked Network Traffic

Taking a closer look at the malware sample itself, there are a number of network-based techniques being used by the malware to detect whether it is being investigated and/or being run within a virtual environment. In the case of modern malware (particularly botnet agents), if the sample cannot reach the Internet and ultimately its command-and-control server, it will typically fail to execute correctly or may act benign.

This is a significant problem for those antivirus vendors and security researchers tasked with determining the maliciousness of the sample being analyzed. As such, a number of games are often played against the malware in order to trick it in to believing it has access to the Internet - and ultimately revealing its malicious functionality and possibly its phone-home credentials. The cat and mouse evolution of detection and evasion of network-based gaming has been going on for several years - particularly as automated malware analysis systems have increasingly been deployed at perimeter inspection points within enterprise networks.

Some of the common techniques used by malware authors to detect the presence of automated analysis environments include the following:

- **"Known Good" Lookup** - The malware attempts to connect to a "known good" Website first to check whether Internet access is possible before attempting to contact the first phone-home

address embedded within itself. The malware will check to see if it can access sites such as [www.google.com](http://www.google.com), [www.yahoo.com](http://www.yahoo.com), [www.cnn.com](http://www.cnn.com), etc. and expect to get a page returned. If a legitimate page is returned, the malware will try to connect to a server controlled by its criminal operator. Upon detection of blocking access to the "known good" lookup, many malware families will simply stop working.

- The malware may also check the response from the "known good" website to retrieve the current date and time. This information is used as a method of verifying that the host (virtualized or otherwise) has not been tampered with and may also be used as a method of synching lists of date-variable command-and-control sites (e.g. the malware may use different servers depending on the current day or time - such as Bobax, Conficker, Sinowal, etc.)
- **"Whitelisted" Downloads** - The initial installation of the malware may be intentionally incomplete (i.e. missing key binaries and DLLs) - meaning that additional malware components must be downloaded from an external site in order for the malware to act maliciously. These sites will often be "whitelisted" legitimate sites that offer free file downloading services (such as [www.filedropper.com](http://www.filedropper.com), [www.rapidshare.com](http://www.rapidshare.com), [www.megaupload.com](http://www.megaupload.com), etc.) and are simply used as public file repositories. If Internet access is blocked, the malware will remain incomplete and no malicious behaviors can be observed (nor can the file be analyzed by other static or heuristic mechanisms).
  - Instead of retrieving binary files from whitelisted file sharing repositories, the malware sample may check public services such as pastebin sites (e.g. [pastebin.com](http://pastebin.com), [pastebin.org](http://pastebin.org), etc.) to retrieve its necessary configuration information - such as a list of command-and-control servers maintained by the criminal operator or a list of malicious actions that malware is expected to perform on installation.
- **Search Result Checking** - In this variant of a "known good" lookup, the malware will attempt to query a popular Internet search engine using a specific keyword (or combination of keywords) with the expectation of retrieving information about the whereabouts of its phone-home server (or other malware configuration information). By using this method, criminal operators can anonymously seed the returned results (without necessarily needing to invest in server infrastructure themselves) and remotely modify configuration settings at any time.
- **Staging Server Check** - The malware agent may initially choose to check an Internet host that is already known (likely to be known) for being under the control of criminals (or exists within a netblock with a poor reputational score) with the expectation of retrieving a specific response from the server. If no answer is forthcoming or an incorrect response is received, the malware will not then attempt to connect to its real command-and-control server (which will have a different address). For automated analysis systems that classify (and block) threats based on observed networked traffic of the malware, these malicious staging checks serve to prevent those systems from identifying the real command-and-control servers - even when network gaming functions are enabled.

Protection technologies that rely on the (automatic) creation of signatures created off the back of observing the correct execution of a captured malware sample are successfully circumvented if Internet access is blocked, filtered or modified.

## Allowed Network Traffic

Malware capable of detecting and responding to the blocking or modification of network traffic is a significant problem for automated malware analysis and protection technologies. Over recent years this kind of functionality has grown in popularity and is a default tactic for several families of malware (and criminal operators) because it is so successful. An obvious way to overcome these kinds of analysis and detection limitations is to not block or modify the network traffic of the malware sample. Unfortunately, even if Internet access is available to the malware sample, additional network-centric techniques can be used by the malware (and criminal operator) to ascertain that it is under automated analysis.

Some of the common techniques used by malware authors to detect the presence of automated analysis environments include the following:

- **Host Signature** - A common way for botnet operators to track successful infections and manage victim machines is to rely on a per-computer unique identifier. These unique identifiers are derived from unique aspects of the victim computer (such as CPU serial number, OS registration key, network card MAC address, etc.) - similar to anti-piracy key generation algorithms used in commercial software - rather than dynamic information such as the IP address, host name or user name. If the botnet operator identifies that the same machine has been infected multiple times or is simultaneously logged in multiple times, they will be able to identify the presence of the malware analysis system.
  - Armed with such knowledge they may choose to add the IP address to a good-guy blacklist, blacklist the specific unique machine identifier and force the malware to appear benign on the analysis system, or prevent the analysis system from being able to download or identify the real command-and-control server(s).
- **Data Signature** - In order for many classes of malware to perform maliciously, they must be exposed to certain stimuli and cached data within the victim's computer (and virtual/emulated/sandboxed analysis environment). The data exposed to the malware agent and subsequently passed to the criminal operator can serve as an identifier of an analysis environment if it is non-unique or appears to be "too random" and therefore faked. Professional botnet operators have developed analysis techniques and tools to aid in this detection as a byproduct of detecting (and evading) honeypot-type technologies
- **Multi-part Malware** - By splitting up the malware into multiple components that must be individually installed on the victims machine prior to behaving maliciously, malware authors can both detect and bypass automated analysis engines. Detection is possible because only some components of the malware will be requested or may be requested in the wrong order (or from the wrong "computer"). Bypasses are also possible because most automated systems do not know how to handle multiple download requests and subsequently subject each binary component to individual examination and evaluation.
- **Desktop Assumptions** - In similar fashion to the "Multi-part Malware" discussed above, the incomplete malware may instead choose to rely on libraries and APIs associated with software known to (or likely to) be installed on victim computers within the target organization. If the automated analysis system does not completely mimic a "real" corporate host - complete with software installations and subsequent data access (e.g. Microsoft Outlook with an Exchange

connector correctly pointing to the mail server with the ability to access the address book) - then the malware will not expose its malicious behaviors and will remain undetected.

- **Locked Malware** - The downloading and installation of a successful malware agent is commonly an iterative process. Many professional criminal organizations dynamically create unique malware samples for each and every victim (i.e. no two malware samples appear the same). One newer technique is to use the unique "Host Signature" (as discussed previously) as a seed for creating each new malware variant and ensuring that the new malware (or any subsequent downloaded component thereof) will not function on any other computer that does not have the same signature. This means that malware samples extracted from victim computers or harvested from streaming network feeds will not execute within an automated analysis system and will therefore not yield any malicious behaviors.

From a corporate enterprise protection perspective, allowing malware samples to be executed within the company's network (even under controlled analysis within dedicated virtual, emulated or sandboxed appliances) carries a significant risk. Things to consider include the following:

1. Enumeration of the analysis technology deployed at the perimeter is highly likely and can be subsequently evaded through tuning of either the malware delivery technique or exploitation of known (public) weaknesses in the technology - thereby making the analysis and protection technology irrelevant. This is a standard tactic for Advanced Persistent Threat (APT) attacks and (more commonly) botnet operators that pursue multiple simultaneous infection campaigns.
2. The malware may attempt to launch a malicious payload upon execution within the analysis environment that is targeted at an external entity. That victim of the attack would attribute the source of the attack to the organization that was executing the malware.
3. The malware, upon detecting that it is operating within an automated analysis system, may choose to exploit known weaknesses within the virtualization technology and compromise the integrity of both the appliance and any other server/device present on the same network segment.
4. Botnet operators are known to specifically target organizations that seek to analyze their malware agents with devastating distributed denial of service (DDoS) attacks. By revealing the presence of such automated (or manual) analysis systems, the organization may be subject to a sustained attack, which would likely affect other critical business systems.

## Conclusions

Modern malware has evolved at a considerable pace in recent years. As more advanced automated analysis techniques have been used by antivirus vendors and smaller-scale implementations have been deployed within the perimeter defenses of enterprise networks, professional malware developers and botnet operators have been forced to implement new methods of detection and adopt new evasion tactics.

Enterprise security teams have found themselves in a "damned if you do, damned if you don't" situation when it comes to implementing in-house automated malware analysis systems. By blocking or modifying Internet communications to the malware samples under investigation they find that their automated

analysis (and detection) systems are easily detected and subsequently evaded. On the other hand, if they allow the malware to communicate over the network or reach their criminal operators they expose their analysis technology (and company) to detection and different forms of evasion.

Wherever possible, enterprise security teams should seek to perform any kind of automated (or manual) malware analysis of captured malware well away from networks and netblocks associated with their organization. They should also plan on changing the physical location or hosting facilities of these analysis systems on a regular basis - monthly or quarterly depending on the volume of malware that needs to be analyzed and the apparent risk of targeted attack against the organization. Ideally, enterprise organizations are encouraged to push their automated malware analysis to cloud-based analysis systems that have no obvious affiliation with their organization.

### **About Damballa, Inc.**

Damballa is a pioneer in the fight against cybercrime. Damballa provides the only network security solution that detects the remote control communication that criminals use to breach networks to steal personal and intellectual information, and conduct espionage or other fraudulent transactions. Patent-pending solutions from Damballa are platform and system-agnostic, protecting networks with any type of device including PCs, Macs, smart phones, as well as mobile and embedded systems. Damballa customers include Fortune 1000 companies, Internet and telecommunications service providers, government agencies and educational organizations. <http://www.damballa.com>

*Copyright © 2011, Damballa, Inc. All rights reserved worldwide*