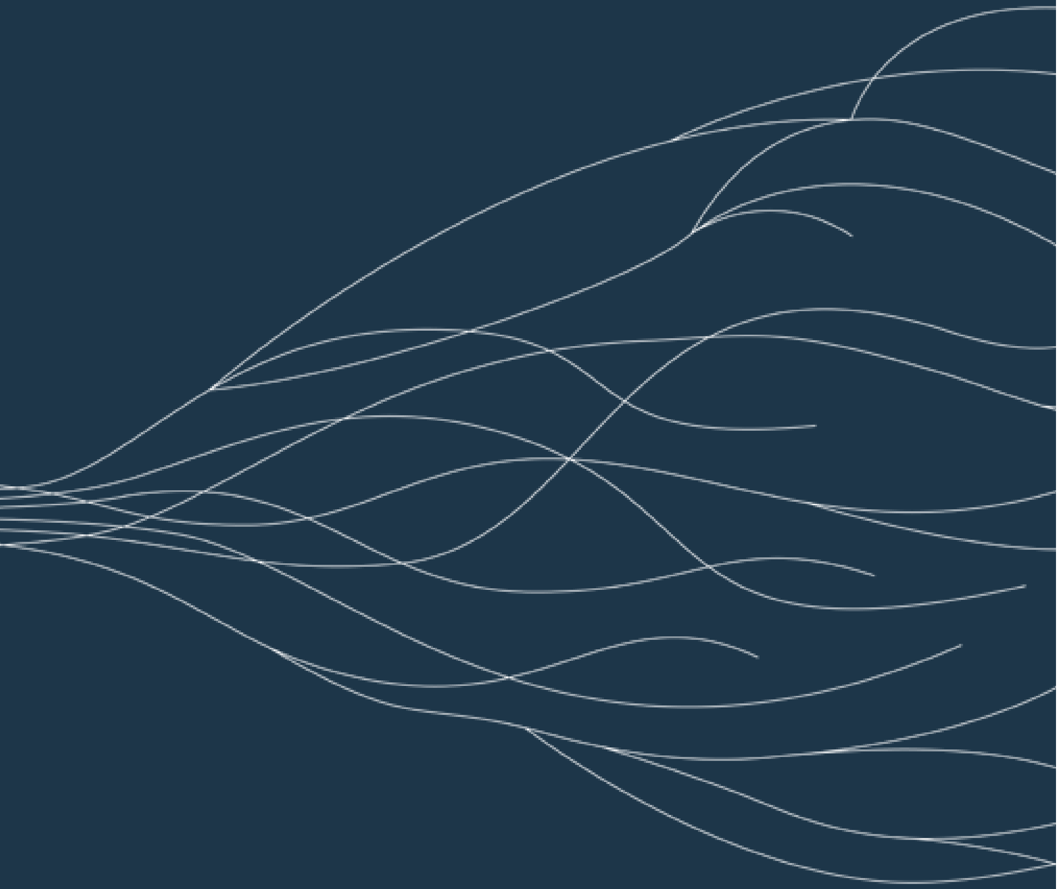
IBM Security



Evading the malware sandbox

How malware avoids detection in research environments

IBM X-Force® Research

Click here to start ▸

◄ Previous Next ►

Understanding advanced malware

Malware in its many forms, from ransomware, viruses and worms to zero-day exploits and bot-network builders, continues to be a popular attack vector among cyber criminals. In a 2016 report by Verizon, 51 percent of breaches analyzed were attributed to malware.[1] Network firewalls, network intrusion detection and prevention devices, and desktop anti-virus software are adept at recognizing and blocking or quarantining known malware. But by their nature, these security measures can only capture what they know.

The challenge comes with malware newly introduced into the wild or advanced malware designed to evade detection and penetrate network security. At that point, the presence of previously unknown code might be flagged by threat monitoring or behavior analysis solutions because of suspicious activity, or a file containing malware may be turned in by a well-trained business user suspicious about an email attachment.

Of course, a full-blown security incident can result when highly skilled attackers use legitimate tools to penetrate networks and then plant unrecognized malware on a workstation or server. This can result from infection by advanced malware engineered to evade the protection capabilities of even the best security solutions in the world—malware introduced by advanced persistent threats (APT) can mutate and remain undetected for weeks or months before executing its perpetrators' orders.

Whether you are investigating an email attachment for potential malware, trying to determine whether or not an unknown file is malicious, or collecting information after the fact of a security incident, it is important to evaluate the behavior of suspicious code. Your goal should also be to collect information that can serve as an indicator of compromise (IOC) that you can share with the security community or use to update your security devices and detect previous infections.

What is advanced malware?

Malware, an abbreviation of “malicious software,” is software created with the intention of doing harm by gaining access to or damaging a computer, typically without the owner being aware. It is an umbrella term that covers a wide range of malicious software, including viruses, worms, Trojans, rootkits, zero-day threats, spyware, adware and ransomware.

Advanced malware, most often used by advanced persistent threat (APT) groups, is malware that is engineered to evade detection for days or months, activating its malicious payload based on some predetermined trigger. It also has advanced capabilities for communication and control, replication or data exfiltration. Advanced malware has a specific mission and may even target a specific organization or specific people within that organization, unlike run-of-the-mill malware that spreads indiscriminately to infect as many people as possible.

Increasingly, advanced malware is designed to evade signature-based protection such as anti-virus software, network protection devices and malware sandbox technology.

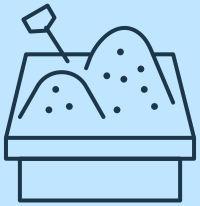
The role of malware sandboxes

Many security organizations employ sandboxing to examine malware in a contained, virtualized environment, where it cannot do any damage. A sandbox is a controlled environment that is set up to look like a regular operating system. Virtual machines are often used to create “runtime sandboxes”[2] that emulate a host computer, complete with operating system and actual hardware—except that the emulation or sandbox software isolates the malware from other system resources, other programs and the network. The malware is then run under supervision, where researchers can examine its deployment routines, payload and malicious activity.

Although sandboxing is a very common way to run malware in a safe zone, legacy malware sandboxes do have their limitations, whether they are proprietary tools, off-the-shelf software or a free online service. Not all have the functionality required to examine the behavior of the malicious code.

Behavior-based malware detection evaluates code based on its intended action. This dynamic evaluation technique works by actually executing the code in the sandbox and watching what happens. However, even execution-based technology may be limited. For example, it may not be capable of evaluating the interaction between malware and the sandbox’s operating system.[3] When the malware is executed for analysis, the malware itself can learn the specific analysis environment, halting all action when it encounters it, or adapting to it to evade that particular sandbox in the future.

The key question is whether or not a legacy sandbox is up to the task of detecting and analyzing advanced malware that is designed specifically to evade detection by sandbox technology.



Advanced malware can be engineered to detect a sandbox environment and employ techniques to avoid analysis.

Sandbox evasion techniques

Malware writers are users of sandbox environments themselves, and as such, they are increasingly adept at developing advanced techniques for evading sandbox technology. The history of evasion goes back to the 1980s, when a piece of malware partially encrypted its own code, rendering the content unreadable by security analysts. Since then a dark market for off-the-shelf evasion technology has developed and is exploited by several contemporary malware families.[4]

That market has been growing, providing attackers with commercial malware packers and "crypting" services to ensure that malware does not get detected on contact with security solutions. Malware itself is also available for sale, with the purveyors advertising that evasive techniques programmed in are part of the "package." Also available on the dark market are digital certificates, sometimes stolen, that are used to sign the malware and make it appear to the operating system to be a trusted executable file. Evasive malware increased 2,000 percent from 2014 to 2015[5], with a sizable portion in 2015 using a combination of 500 techniques designed to avoid detection and analysis.[6] It's reasonable to assume that the number of techniques discovered and made available to malware writers have increased significantly since then.

This section describes the more common sandbox evasion techniques used by malware. Many of these techniques are used in parallel to make it even more difficult and time consuming to detect and disable malware infections.

Environmental awareness

Malware can be programmed to determine whether or not it is trying to infect a runtime sandbox. The presence of VMware registries or running processes are often signals that reveal the sandbox. If the malware detects that it is in a virtualized or a "bare metal" environment, it will execute only benign behavior, or not execute at all, nor fetch other modules or configurations—and thus it can't be analyzed.

For example, URLZone, also known as Bebloh/Shiotob, is a sophisticated banking Trojan with extensive anti-research features for detecting a sandbox/research environment, whereby it does not execute. The malware employs process hollowing to detect devices installed on the host environment, looking for indicators of VMs (such as VMWare). URLZone also checks its own file name for substrings like "Sample," "Virus" and "Sandbox" before executing. And it uses timing checks to determine the presence of a hypervisor call, which takes significantly longer to execute a calculation than a physical machine would.

GootKit, one of the more advanced banking Trojans in operation, has high walls built around it to prevent access by researchers. Even before the malware is installed, GootKit's dropper verifies the system's processor values inside the Windows Registry by searching for a specific name used in servers. The dropper also looks for VM resources on disk, checks the device's BIOS to find values that may indicate a virtual machine client installation, and examines the machine's MAC address.

If all is clear, GootKit's true payload is fetched and deployed, and the malware itself performs additional VM checks. It checks for a whitelist of acceptable CPU names that indicate a physical endpoint or, alternatively, a VM. Next it scans for hard drives on the target machine, looking for VM values that would be difficult for researchers to disguise.

Human interaction detection

Malware can also be programmed to detect user interaction. The wrong kind of user interaction—or the lack thereof—is interpreted as an indicator that the malware is operating in a sandbox. Common interactions that the malware might look for include human-like scrolling—specifically, scrolling to a specific point in a file—multiple mouse clicks, or mouse speed that is not suspiciously fast.[7]

System interaction detection

To deploy undetected, malware developers also work security evasion into the execution routine. For example, the Client Maximus malware uses a stealthy driver to scan for a list of anti-virus programs, looking for active processes on the OS. If it finds any, it shuts them down before executing the malware's payload.

Another malware, the Zeus Maple variation of the Zeus v2 Trojan, uses a loader that begins by denying access to its code to any running security processes. To prepare the environment for the eventual payload execution, it further uses an interesting twist on what's known as the self-debugging technique to evade code examination on the Windows OS. It manipulates Microsoft's PAGE_GUARD page protection modifier, making the flag permanent before executing the malware. This way, the loader thwarts the attempts of security software and research tools to examine or interact with its code.

Timing-based evasion

There are a number of timing-based techniques in use to evade detection. These can work because sandbox resources are precious, and thus the analysis system can only spend a limited amount of time on each test run of suspicious code. One technique is "stalling," when the malware executes useless CPU cycles disguised to look like non-malicious activity,[8] but in reality, delaying the actual nefarious code until the sandbox has determined innocent activity and passes the malicious file into the environment.

Another technique is illustrated by the QakBot Trojan, which is ushered into infected endpoints through a dropper. The dropper typically uses delayed execution to evade detection. It lands on the target endpoint and halts, waiting 10 to 15 minutes before any further action, hoping to elude sandboxes that might try to analyze in upon arrival.

In other timing-based invasions, the malware is programmed to run only at certain times, like within a limited amount of time since its compilation by the attacker, waiting for a specific time and date to execute, or following certain user actions, such as opening a browser after initial infection and waiting for the user to click, or activating only after the system reboots.

Obfuscating internal data

Another common evasion technique malware developers often use is obfuscating internal data. Malware authors have a variety of tricks at their disposal, including encrypting strings and communications, or concealing some malware code by replacing program data with hashed values so the code can't be read by traditional sandboxes.[9] The infamous banking Trojan Dridex, for example, focuses on encrypting API calls, going so far as to introduce a new API obfuscation method in 2016.

In a similar fashion, malware can use obfuscation techniques to evade communication interception by security research or hostile parties. The Necurs botnet does this with a domain generation algorithm (DGA) that can generate 2,048 domain names for command-and-control (C&C) servers, covering over 43 different top-level domains, that die out every four days. Using a DGA is common for advanced malware, keeping outsiders guessing as to which domain the malware will fetch its updates and instructions from, and preventing the shutdown of those pivotal resources. The Andromeda botnet uses several keys to encrypt data for communications with C&C servers to make tracking and interception difficult. Additionally, the malware has anti-sandboxing built in to prevent security researchers from analyzing it.

Out-maneuvering malware with an advanced sandbox

One alternative to buying, managing and updating an internal malware sandbox is IBM® X-Force® Malware Analysis on Cloud. This software as a service (SaaS) malware sandbox solution helps security analysts identify malware using rigorous behavior-based analysis that is designed to help outsmart sandbox-evading technology.

IBM X-Force Malware Analysis is integrated with both the X-Force Exchange threat intelligence platform and with solutions in the IBM Security portfolio. Malware can be submitted for analysis either online via the X-Force Exchange or directly from select IBM Security solutions.

The first step in the analysis is to filter submissions for known bad files, returning results nearly immediately. The service matches the sample against known bad file hashes in an existing library of millions of samples and then will look for malformed code, file type, bad file structure and obfuscated code—all indicators that the file submitted is not to be trusted.

Dynamic behavioral analysis

Files that are not identified as “known bad” in the filtering process are submitted to the cloud-based sandbox for further analysis. This behavioral analysis sandbox solution offers features designed to combat anti-sandbox code that can often evade detection by legacy sandboxes. The cloud-based infrastructure that runs X-Force Malware Analysis provides a sandbox environment that can scale as required to analyze executable content across multiple platforms.

The sandbox interacts with the malware itself to elicit possible execution paths. It can dynamically create environmental information and emulate process interactions that make the sandbox appear to be a host computer. It can also alter malicious code execution to bypass anti-sandbox strategies for evading detection.

◄ Previous Next ►

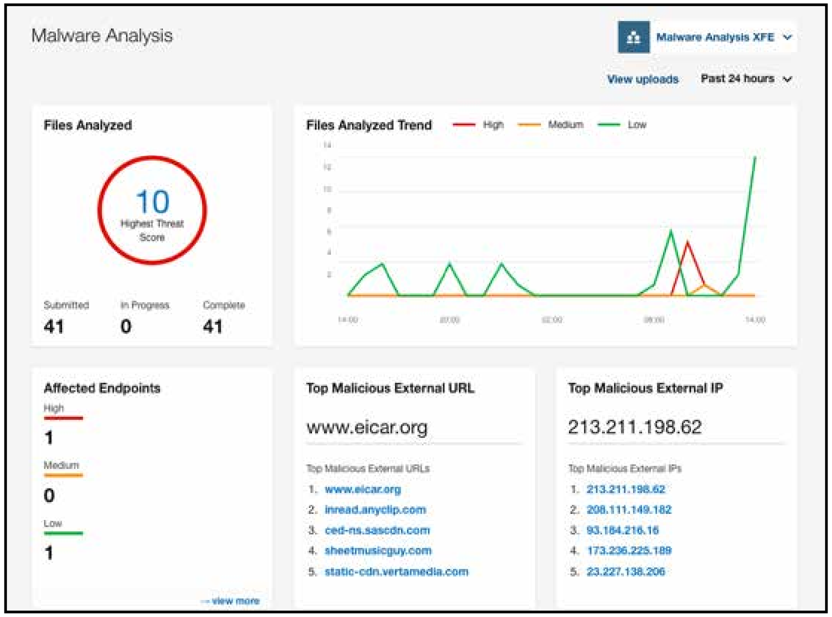
Contents

Scoring and reporting

Suspicious files receive one of three ratings, based on a score of 0 – 100: Benign (for 1 – 29 points), Nuisance (for 30 – 69 points) or Malicious (for 70 – 100 points). The scoring is comprised of five categories:

1. Malware actions or activity: what behaviors is it exhibiting?
2. Reputation data: is it hosted on a suspicious IP or URL?
3. Digital certificate: is the file correctly signed?
4. VirusTotal: is this a known sample?
5. External reputation: is the file a popular app?

Clear reports available via an online dashboard enable users to keep track of files submitted, screening or analysis results and the number of endpoints affected. Detailed reports for files that undergo sandbox analysis are also available online.



Files Analyzed

10 Highest Threat Score

Submitted	In Progress	Complete
41	0	41

Affected Endpoints

High 1

Medium 0

Low 1

Top Malicious External URL

www.eicar.org

Top Malicious External URLs

1. www.eicar.org
2. inread.anyclip.com
3. ced-ns.sascdn.com
4. sheetmusicguy.com
5. static-cdn.vertamedia.com

Top Malicious External IP

213.211.198.62

Top Malicious External IPs

1. 213.211.198.62
2. 208.111.149.182
3. 93.184.216.16
4. 173.236.225.189
5. 23.227.138.206

Figure 1. Example of the online dashboard from the IBM X-Force Malware Analysis on Cloud service

◄ Previous Next ►

Contents

About IBM X-Force

The IBM X-Force Research studies and monitors the latest threat trends including vulnerabilities, exploits, active attacks, viruses and other malware, spam, phishing, and malicious web content. In addition to advising clients and the general public about emerging and critical threats, IBM X-Force also delivers security content to help protect IBM clients from these threats. Threat intelligence content is delivered directly via the IBM X-Force Exchange collaborative platform, available at xforce.ibmcloud.com

For more information

To learn more about IBM X-Force Malware Analysis on Cloud, talk to your IBM representative or IBM Business Partner, or visit: **ibm.com**/security/xforce

To register to try X-Force Malware Analysis on Cloud free for 30 days with your IBM ID, visit: **ibm.com**/us-en/marketplace/malware-analysis

Follow @IBMSecurity on Twitter or visit the IBM Security Intelligence blog

[1] 2017 Data Breach Investigations Report, Verizon

[2] https://www.techopedia.com/definition/25266/sandboxing

[3] https://www.infosecurity-magazine.com/opinions/malware-detection-signatures/

[4] https://betanews.com/2017/06/20/malware-evasion-techniques/

[5] https://www.rsaconference.com/events/us15/agenda/sessions/2022/evasive-malware-exposed-and-deconstructed

[6] https://www.tripwire.com/state-of-security/security-data-protection/the-four-most-common-evasive-techniques-used-by-malware/

[7] https://www.secureworks.com/blog/bg-5-ways-malware-evades-the-sandbox

[8] https://www.secureworks.com/blog/bg-5-ways-malware-evades-the-sandbox

[9] https://www.tripwire.com/state-of-security/security-data-protection/the-four-most-common-evasive-techniques-used-by-malware/

◄ Previous Next ►

IBM Security

IBM Security
75 Binney Street
Cambridge MA 02142

Produced in the United States of America
October 2017

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.