International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1218

(10/2020)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Cyberspace security – Cybersecurity

Requirements and guidelines for dynamic malware analysis in a sandbox environment

Recommendation ITU-T X.1218

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES (1)	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols (1)	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	**X.1200–X.1229**
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES (2)	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1319
Smart grid security	X.1330–X.1339
Certified mail	X.1340–X.1349
Internet of things (IoT) security	X.1360–X.1369
Intelligent transportation system (ITS) security	X.1370–X.1389
Distributed ledger technology security	X.1400–X.1429
Distributed ledger technology security	X.1430–X.1449
Security protocols (2)	X.1450–X.1459
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699
QUANTUM COMMUNICATION	
Terminologies	X.1700–X.1701
Quantum random number generator	X.1702–X.1709
Framework of QKDN security	X.1710–X.1711
Security design for QKDN	X.1712–X.1719
Security techniques for QKDN	X.1720–X.1729
DATA SECURITY	
Big Data Security	X.1750–X.1759
5G SECURITY	X.1800–X.1819

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1218

Requirements and guidelines for dynamic malware analysis in a sandbox environment

Summary

Unknown malware is commonly used in advanced attacks, in particular advanced persistent threats (APTs), to avoid being detected. For example, a targeted attack using phishing email weaponized with unknown malwares can easily achieve a successful initial compromise. Thus, for detection of advanced attacks, special attention and defence measurements should be taken to detect unknown malwares. Recommendation ITUT-T X.1218 analyses threats related to unknown malwares and specifies requirements of unknown malware detection based on dynamic behaviour analysis.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1218	2020-10-29	17	11.1002/1000/14444

Keywords

APT, dynamic behaviour analysis, sandbox, unknown malware.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.1218

Requirements and guidelines for dynamic malware analysis in a sandbox environment

1 Scope

This Recommendation analyses threats related to unknown malwares and specifies requirements of unknown malware detection based on dynamic behaviour analysis.

This Recommendation covers:

– Analysis of threat scenarios related to unknown malwares, such as targeted/customized malwares; and

– Requirements of unknown malware detection based on dynamic behaviour analysis in a sandbox environment.

This Recommendation does not analyse the content of the detected object but only its executing related behaviours, for example calls of system level application programming interfaces (APIs), etc.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following term defined elsewhere:

3.1.1 malware [b-ITU-T X.1211]: Malicious software designed specifically to damage or disrupt a system, attacking confidentiality, integrity and/or availability.

NOTE – Examples include: viruses, ransomware, spyware, adware and scareware.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 zero-day exploit: A cyber-attack that occurs on the same day a weakness is discovered in software.

3.2.2 phishing: A means by an individual or group to solicit personal information or to achieve initial intrusion by unsuspecting users employing social engineering techniques.

3.2.3 phishing emails: Phishing emails are crafted and tend to appear as sent from a legitimate organization or known individual. They attempt to entice the receiver users to click on a link that will take the user to a malicious web page or to open a malicious attachment.

3.2.4 sandbox: Sandbox is a simulated secure environment which is separated from the underlying host machine/operating system, in which programs/samples can be executed, and the dynamic process behaviours of the sample can be observed and thus the hidden malicious codes can be detected.

3.2.5 unknown malware: Unknown malware is malware that has not been discovered yet, which means that no signatures or features exist for it in most security systems such as antivirus or antispyware software. There is always a gap period between the time a malware starts spreading across the Internet and the time its signature or feature is available. During this gap period the malware is an unknown malware.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

HTML Hyper Text Markup Language

ICT information and communication technology

IDS Intrusion Detection System

IM Instant Messaging

JS JavaScript

ML Machine Learning

PE Portable Executable

URL Uniform Resource Locator

VM Virtual Machine

XML extensible Markup Language

5 Conventions

In this Recommendation:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "**is prohibited from**" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "**can optionally**" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Threat scenarios

6.1 Targeted/customized malware threat

Cyber threats are evolving along with the development of information and communication technologies (ICTs), especially for example, targeted malware is growing in popularity. More and more, hackers are creating completely new malware for individual targets, using vulnerabilities and

infrastructure weaknesses of the targeted system for more effective attacks. As most of the traditional defence products are signature or feature based, there is a lack of detection capabilities for unknown threat attacks.

6.2 Phishing email threat

In order to avoid being detected by multi-level security devices (such as firewalls, intrusion detection systems (IDS), etc.) and enable the attack to directly reach the target network, attackers often prepare a malware and hide it in a file as an email attachment, or prepare a phishing uniform resource locator (URL) and then deliver the email with the attachment or the URL to targeted users, confusing and inducing the target to click to release the attack.

6.3 Instant messaging propagation threat

Attackers may send disguised malicious files (such as disguised photos, tools, orders, etc.) directly to the victim through social software such as ICQ, QQ, WeChat or other instant messaging (IM) tools to trick the victim into clicking.

6.4 Web browsing or downloading threat

Website malicious code: Attackers may build attack servers, put malwares on a server in the form of URL links or JavaScript (JS) scripts, and induce victims to download and execute by deception.

Water-hole attack: The attacker obtains the target's habit of browsing or using a network service by social engineering or other means. Then the attacker exploits these websites or servers and attaches malwares there. When the victim visits the above websites or services, he/she may be attacked.

6.5 Supply chain attack threat

By attacking some very popular or widely used version-upgrade servers of devices/systems/software/services, the attacker directly pushes the system versions or patches containing malicious code to the victim through the version-upgrade server. Manipulating common software development tools leads software developers to download the tools containing malicious code.

6.6 Ransomware threat

Ransomware is a kind of malicious software that infects computer systems, restricts access to the victim's data and requires a ransom. Because access to the computer is blocked, the victim will be forced to pay the person who developed the malicious program in order to remove the restriction. Ransomware attacks are typically carried out using a trojan that is disguised as a legitimate file that the user is tricked into downloading or opening when it arrives as an email attachment.

For example, WannaCry ransomware began affecting computers worldwide and propagated automatically between computers without user interaction. Unlike ordinary ransomware, which is spread via email attachments, computers were infected just by accessing the Internet network. WannaCry ransomware encrypts various types of files such as document files, compressed files, database files and virtual machine files.

6.7 Fileless malware threat

Fileless malware is a type of malicious software that uses legitimate programs to infect a computer. It is a malicious code that does not actually have a file and operates in memory by inserting malicious code into normally trusted processes to execute an attack. It does not rely on files and leaves no footprint in the infected machine. Instead, it takes advantage of existing vulnerabilities on the victim's machine.

There are many techniques that attackers might use to launch a fileless malware attack. An example scenario could be a target machine that receives malicious network packets that exploit a certain vulnerability which leads to the installation of a backdoor, and ends up residing only in the kernel memory. In this case, there is no file, nor any data written on a file.

7 Sandbox technology

Sandbox technology is used to safely execute suspicious code without risking harm to the host device or network. It can avoid system failures and keep software vulnerabilities from spreading.

Traditional security measures are reactive and based on signature detection, which works by looking for patterns identified in known instances of malware. Because only previously identified threats can be detected, sandbox is an important security equipment especially for unknown malware detection. Although the malicious codes and attacking methods are easy to change, the aggressive behaviours are relatively fixed. Therefore, dynamic behaviour analysis in a sandbox environment is an effective way to cope with unknown malwares, and there is still a need to complement these sandbox technologies with an advanced unknown malware detection.

8 General requirements and guidelines on sandbox

For dynamic malware analysis in a sandbox environment, the following requirements are needed:

8.1 Comprehensive analysis environment

The types of files commonly used by malwares in email attacks are: Office, Archive, Pdf, Exe, Binaries, XML/HTML/JS, IOS app, image, Scripts, Android apk, etc. The sandbox should support an operating system image environment that is adapted to different malware types in order to trigger the malware to start running. The sandbox should support as many file types as possible and be able to identify file types in order to select the appropriate software to run the file automatically. This software should be pre-installed in images, and different software versions are required, as many malwares are designed for the vulnerability of a particular software version. After the sandbox receives the sample, it can identify the file type by pre-parsing the file or using some static scanning tools such as YARA, and then open the correct version of software to support that file type.

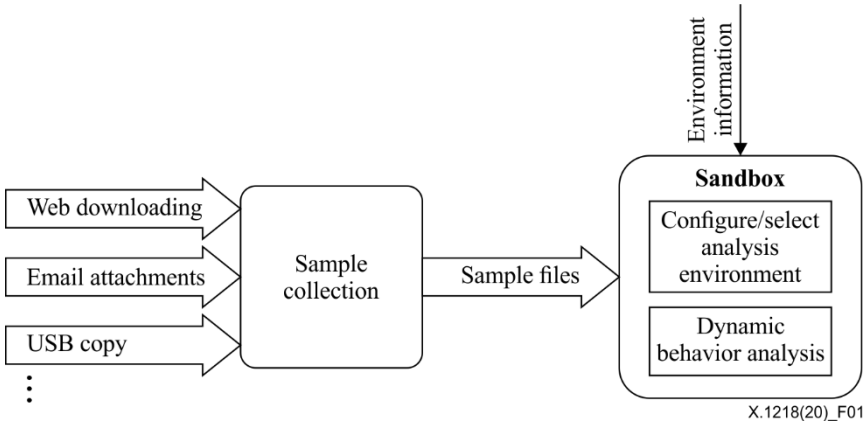

Figure 1 – Sample analysis

In order to improve the effectiveness of sample analysis, especially for targeted attack, it is necessary to collect the sample file and obtain the environment information corresponding to the sample file, then configuring or selecting a matching image environment according to the environment information, and analysing the dynamic behaviour of the sample file in the image environment as shown in Figure 1.

8.2 Identification of disguised file types

Malwares often disguise file types, for example, a PE file is disguised with an extension .abc. Sandboxes should be able to identify disguised file types and run it appropriately.

8.3 Isolation from the outside

No matter what kind of virtualization technology is used, software/hardware simulation technology or other sandbox technology, sandboxes should be isolated from the external system to avoid attacks from outside and prevent the impact from the detected samples in the sandbox.

8.4 Resistance to evasion

If a sandbox does not perform monitoring in a stealthy way, there is a good chance malware will recognize it is under surveillance and alter its behaviour to "play nice". In this case the malware could show non-suspicious behaviour or just stop running.

Malware creators have become quite knowledgeable about sandboxing software and have developed sophisticated ways to detect the simulated environment and to evade detection. One example is the stalling technique, which executes "useless" computations to give the appearance of normal activity until the sandbox times out. Once the sandbox times out it unleashes the malicious code onto the system. The sandbox could support system acceleration to prevent malware from evading detection through silence. For sandboxes in virtual machine (VM) form, one can start the time-varying mode of the VM, and obtain the record log of the tested sample in the time-varying mode of the VM, then analyse whether there is malicious behaviour in the record log under this mode. If there is malicious behaviour in the record log, the latent behaviour type in time dimension of the tested sample will be determined according to the type of the time-varying mode.

8.5 Separate Internet access

The sandbox system could be able to connect to the outside network through a separate network channel to prevent malicious samples from requesting through network connection as a decision switch for the next malicious steps or for downloading other malware.

8.6 Scalability of analysis and management

In addition to focusing on the detection capabilities of the sandbox system, the scalability of the system should also be taken seriously. The products should be able to handle high volumes of traffic and analyse objects/files.

9 Further requirements and guidelines on dynamic behaviour analysis

9.1 High level of visibility into malware behaviour

When monitoring the behaviour of a file, almost all sandboxes look at the system call interface or the Windows API. System calls are functions that the operating system exposes to user mode processes, so that they can interact with their environment (e.g., read files, send packets over the network, read a registry entry on Windows, etc.). System calls and Windows API function calls need to be monitored but this is only one piece of the puzzle. A sandbox that only monitors system calls is blind to everything that happens in between. Thus, a sandbox should support the capture of various behavioural granularities, including call detection of system level APIs and machine level privileged instructions.

9.2 Behaviour granularity capture and analyse

Malicious behaviour of malwares can be triggered in different ways, not only by API calling, but also by system interruption or specific machine instructions. Dynamic analysis sandbox should be able to

capture the above granularity behaviours, i.e., API calls, assembly language calls to system interrupts, machine-specific instructions.

Malwares may execute by special jump, call or return instructions, so they can be identified by counting the number and order of instructions executed. Thus, statistical analysis of machine instructions and identification of abnormal sequences are examples that can be used for further analysis.

9.3 Record of API calls

The API calls and input parameters, execution results and execution time of API of a sample should be recorded. The recorded API name, input parameters, execution results and execution time, execution order and number of times can be used to judge whether the file is malicious or not and determine the malicious behaviours of the checked sample.

9.4 Action abstraction

Actions should be abstracted from original API records rather than just taking raw logs. Examples of actions includes adding start-up entries by creating/modifying registries, creating and writing portable executable (PE) files to create other malicious files, downloading other attack files through the Internet, etc.

9.5 Ability to decompress compressed files

An attacker usually uses compressed malicious files and then transmits them to the victim. Thus, analysis of compressed files should be supported. First, the vulnerability of the decompression software (such as CVE-2018-20250) should be analysed, and then every decompressed file should be analysed.

9.6 Analysis of derivative files

It should be possible to analyse files released or further downloaded through the network from the original detected sample. If a new file is started by the current process as another process or opened or loaded by the current process, the new process or loading process should also be recorded. If the new file is not eventually executed by the current process, a separate analysis project for the new file is then needed.

9.7 Threat determination

The sandbox should be able to define threat types and levels either based on rule matching or by other means such as through machine learning (ML).

Bibliography

[b-ITU-T X.1211] Recommendation ITU-T X.1211 (2014), *Techniques for preventing web-based attacks*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020