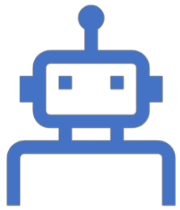


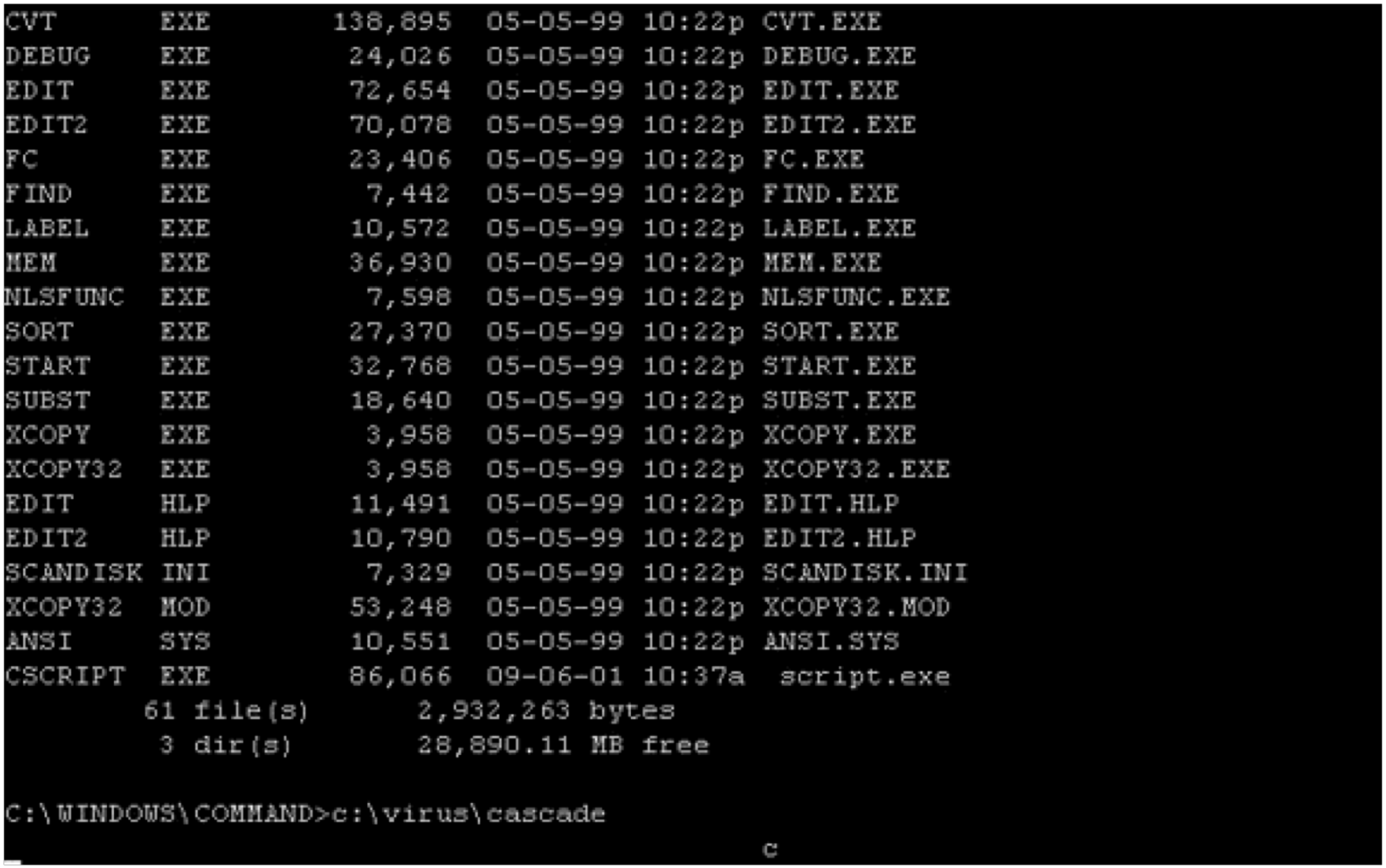
Malware Analysis

Corrado Aaron Visaggio

Associate Professor

Dept. Of Engineering

University of Sannio


CVT EXE 138,895 05-05-99 10:22p CVT.EXE
DEBUG EXE 24,026 05-05-99 10:22p DEBUG.EXE
EDIT EXE 72,654 05-05-99 10:22p EDIT.EXE
EDIT2 EXE 70,078 05-05-99 10:22p EDIT2.EXE
FC EXE 23,406 05-05-99 10:22p FC.EXE
FIND EXE 7,442 05-05-99 10:22p FIND.EXE
LABEL EXE 10,572 05-05-99 10:22p LABEL.EXE
MEM EXE 36,930 05-05-99 10:22p MEM.EXE
NLSFUNC EXE 7,598 05-05-99 10:22p NLSFUNC.EXE
SORT EXE 27,370 05-05-99 10:22p SORT.EXE
START EXE 32,768 05-05-99 10:22p START.EXE
SUBST EXE 18,640 05-05-99 10:22p SUBST.EXE
XCOPY EXE 3,958 05-05-99 10:22p XCOPY.EXE
XCOPY32 EXE 3,958 05-05-99 10:22p XCOPY32.EXE
EDIT HLP 11,491 05-05-99 10:22p EDIT.HLP
EDIT2 HLP 10,790 05-05-99 10:22p EDIT2.HLP
SCANDISK INI 7,329 05-05-99 10:22p SCANDISK.INI
XCOPY32 MOD 53,248 05-05-99 10:22p XCOPY32.MOD
ANSI SYS 10,551 05-05-99 10:22p ANSI.SYS
CSCRIPT EXE 86,066 09-06-01 10:37a script.exe
61 file(s) 2,932,263 bytes
3 dir(s) 28,890.11 MB free
C:\WINDOWS\COMMAND>c:\virus\cascade
c


BAD RABBIT
If you access this page your computer has been encrypted.
Time left before the price goes up
41:18:14
Price for decryption:
- 0.05
Enter your personal key or your bitcoin address

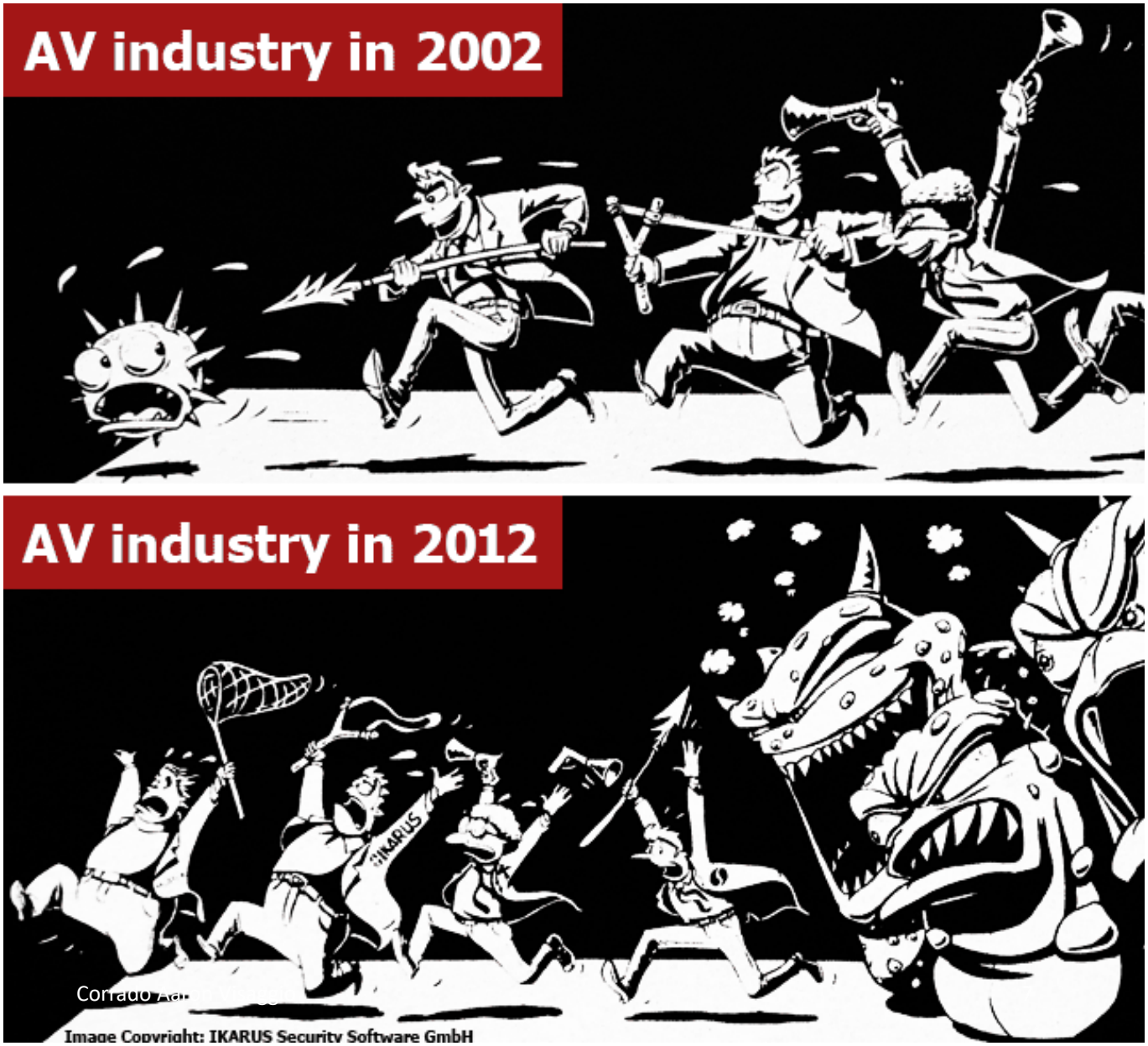
Come ci arrivavano i virus…una volta

Come ci arrivano oggi i virus

John McAfee: "Antivirus is dead"

by OWEN WILLIAMS — Sep 3, 2015 in INSIDER

AV vs Malware



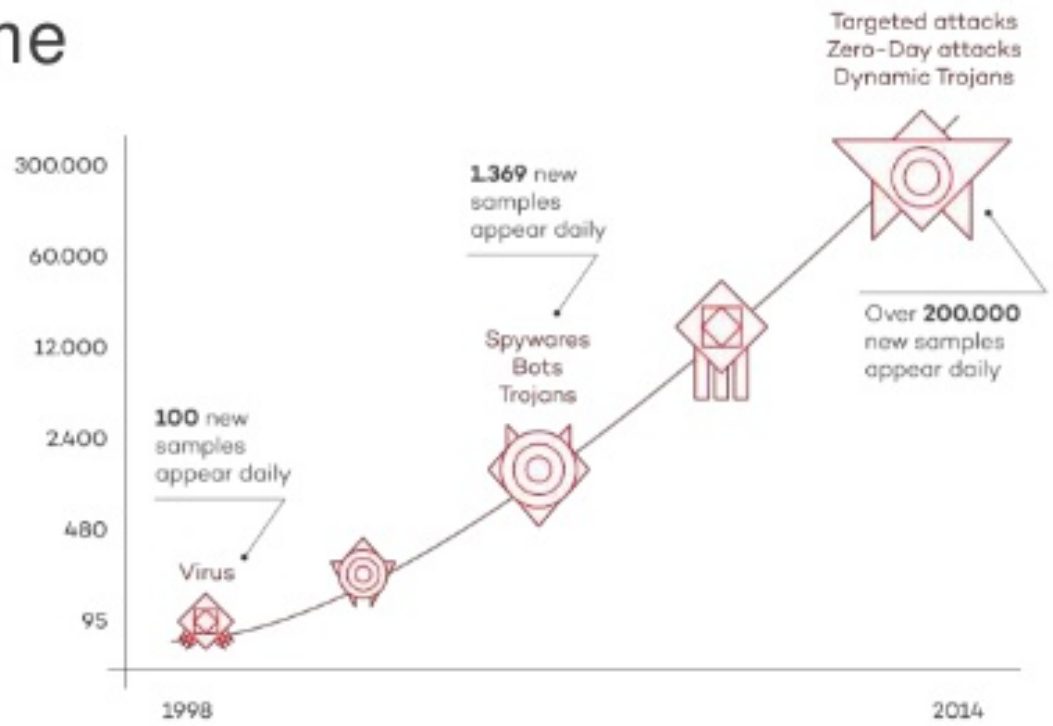
UNIVERSITÀ DEGLI STUDI DEL SANNIO Benevento

The best Antivirus

Corrado Aaron Visaggio

UNIVERSITÀ DEGLI STUDI DEL SANNIO Benevento

Malware volume evolution

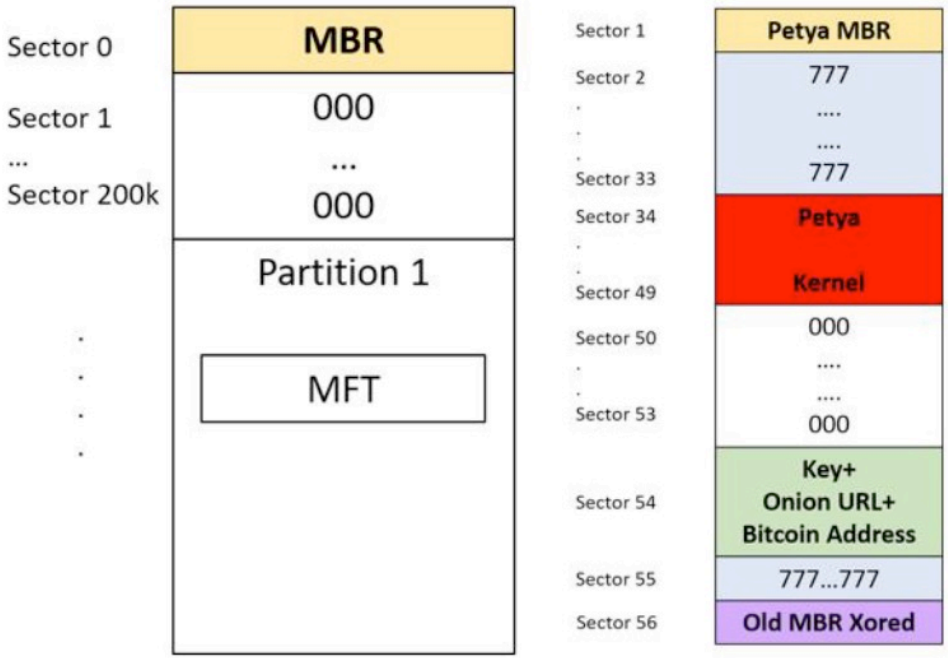


Malware: the bare facts

- **200,000** malware a day
- **80M** new malware a year
- **700M+** malware in the wild
- **Targeted malware** vs automatic generated malware
- **Zero-day** e conceived cyber-arsenals

Petya

- encrypts the **MFT** (Master File Table) of the **NTFS** filesystem **preventing Windows from booting**
- The **old MBR** is not deleted, but **encrypted** by xoring with the key "7" and **moved** in the sector 56.
- Replaces the **MBR** with the **Petya Bootloader**
- "*NtRaiseHardError()*" -> crash the system in user mode -> reboot -> the machine executes **Petya MBR**



NotPetya – the evolution after a couple of months

- What's new:

1. It **cyphers** some **user files before** the **reboot** of the machine
2. It uses the famous and devastating **Eternalblue exploit**, based on a vulnerability of SMB Windows protocol (MS17-010; CVE-2017-0143) -> **worm behavior**.
3. It **schedules** a **legal reboot** instead of forcing it
4. It presents a **different user interface** after the reboot.

Agenda

Introduction to malware

Malware Analysis

Malware as a commodity

Perspectives on Malware

An introduction to Malware


ILOVEYOU - Message (Rich Text)
File Edit View Insert Format Tools Actions Table Help
Reply
Reply to All
Forward
From: John Doe
Sent: Thu 5/4/00 11:29 AM
To: John Doe
Cc:
Subject: ILOVEYOU
kindly check the attached LOVELETTER
LOVE-LETTER-FOR-YOU.TXT.vbs
coming from me.

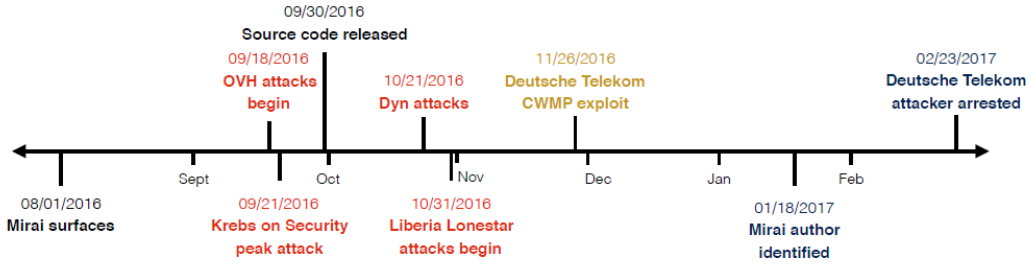
The new menaces (arent’ they cute?)

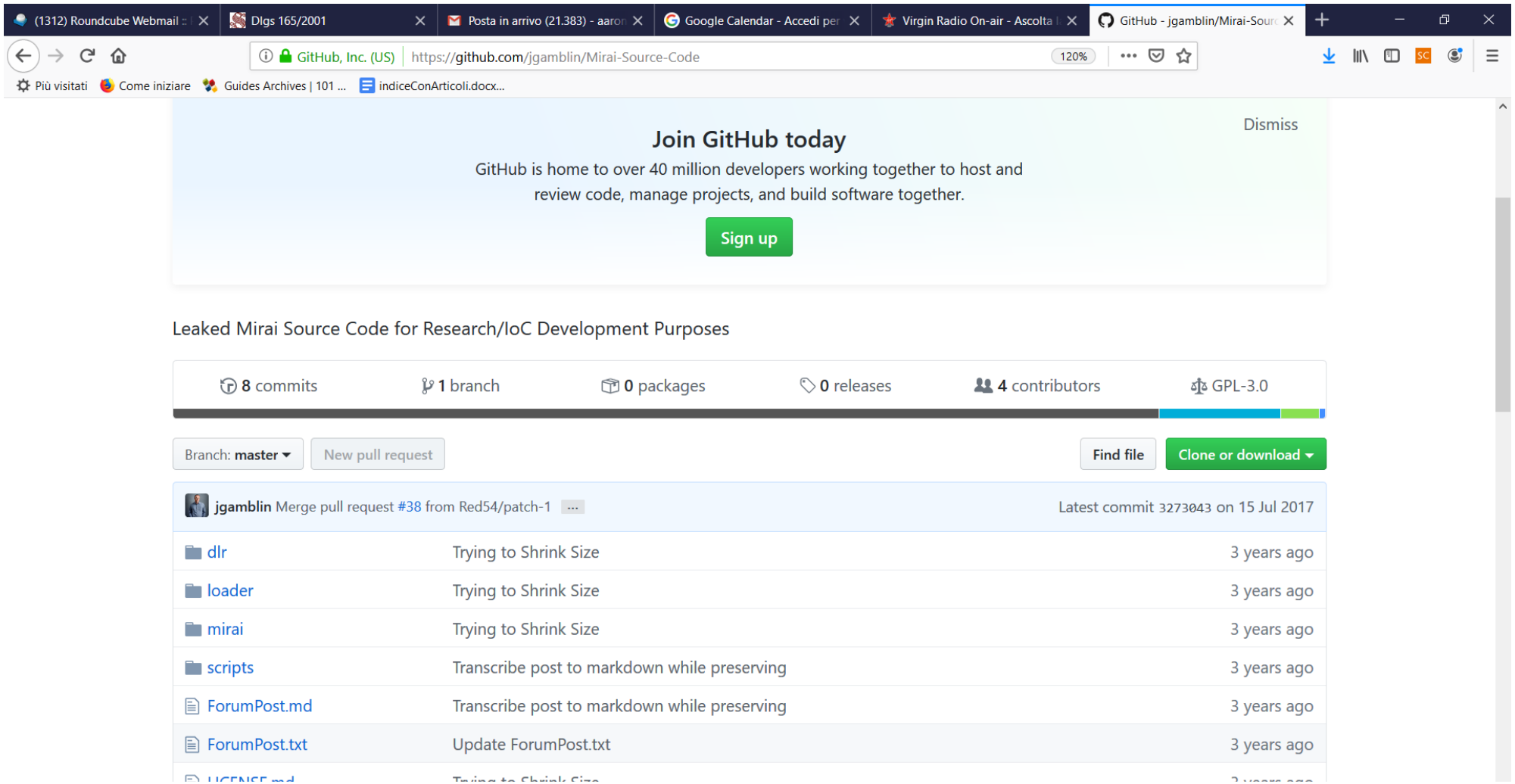
- Fileless
- Aggressive Evasion techniques
- Implants
- GAN (?)

MIRAI

- In September 2016, a spree of massive distributed denial-of-service (DDoS) attacks temporarily crippled Krebs on Security, OVH, and Dyn.
- attack on Krebs exceeded 600 Gbps in volume -> 10^5 compromised devices
- Success factors:
  - efficient spreading based on Internet-wide scanning,
  - rampant use of insecure default passwords in IoT products, and
  - keeping the botnet's behavior simple would allow it to infect many heterogeneous devices
- the botnet infected nearly 65,000 IoT devices in its first 20 hours before reaching a steady state population of 200,000–300,000 infections
- These bots fell into a narrow band of geographic regions and autonomous systems, with Brazil, Columbia, and Vietnam disproportionately accounting for 41.5% of infections.

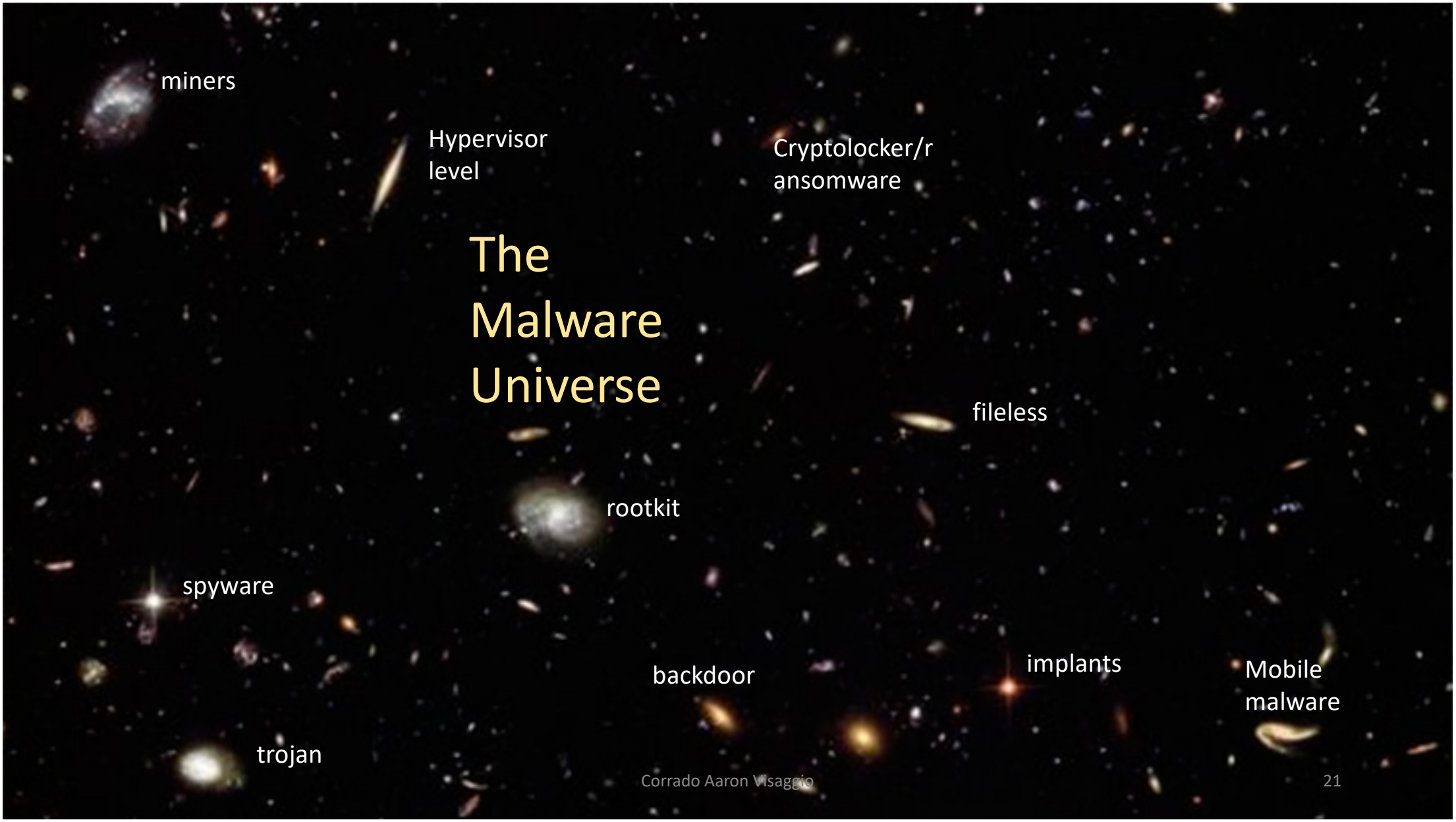
MIRAI timeline



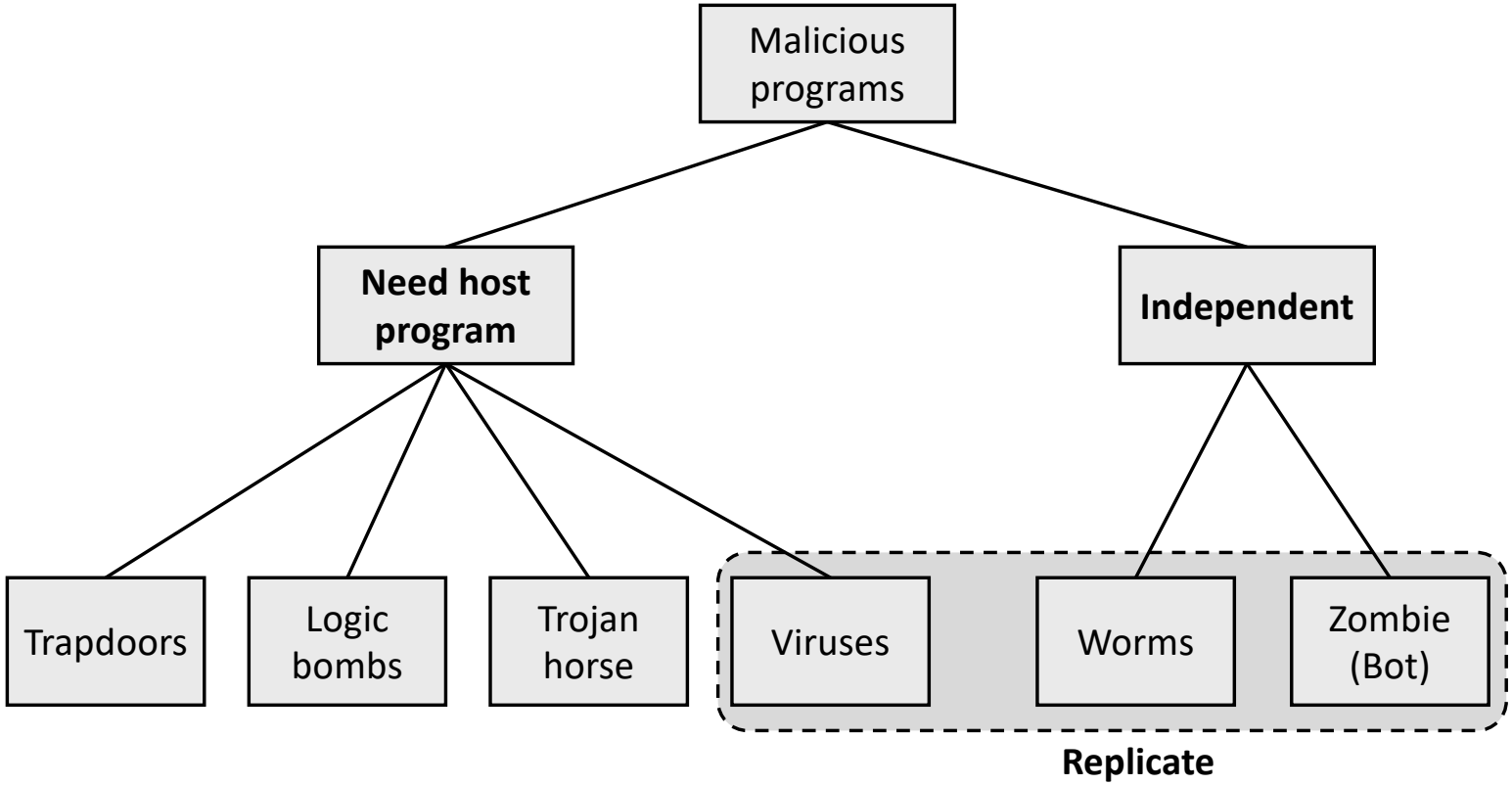

(1312) Roundcube Webmail
DIgs 165/2001
Posta in arrivo (21.383) - aaron
Google Calendar - Accedi per
Virgin Radio On-air - Ascolta
GitHub - jgamblin/Mirai-Sour
GitHub, Inc. (US)
https://github.com/jgamblin/Mirai-Source-Code
120%
Più visitati
Come iniziare
Guides Archives | 101 ...
indiceConArticoli.docx...
Dismiss
Join GitHub today
GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.
Sign up
Leaked Mirai Source Code for Research/IoC Development Purposes
8 commits
1 branch
0 packages
0 releases
4 contributors
GPL-3.0
Branch: master
New pull request
Find file
Clone or download
jgamblin Merge pull request #38 from Red54/patch-1
Latest commit 3273043 on 15 Jul 2017
dlr Trying to Shrink Size 3 years ago
loader Trying to Shrink Size 3 years ago
mirai Trying to Shrink Size 3 years ago
scripts Transcribe post to markdown while preserving 3 years ago
ForumPost.md Transcribe post to markdown while preserving 3 years ago
ForumPost.txt Update ForumPost.txt 3 years ago

Is malware that ugly?

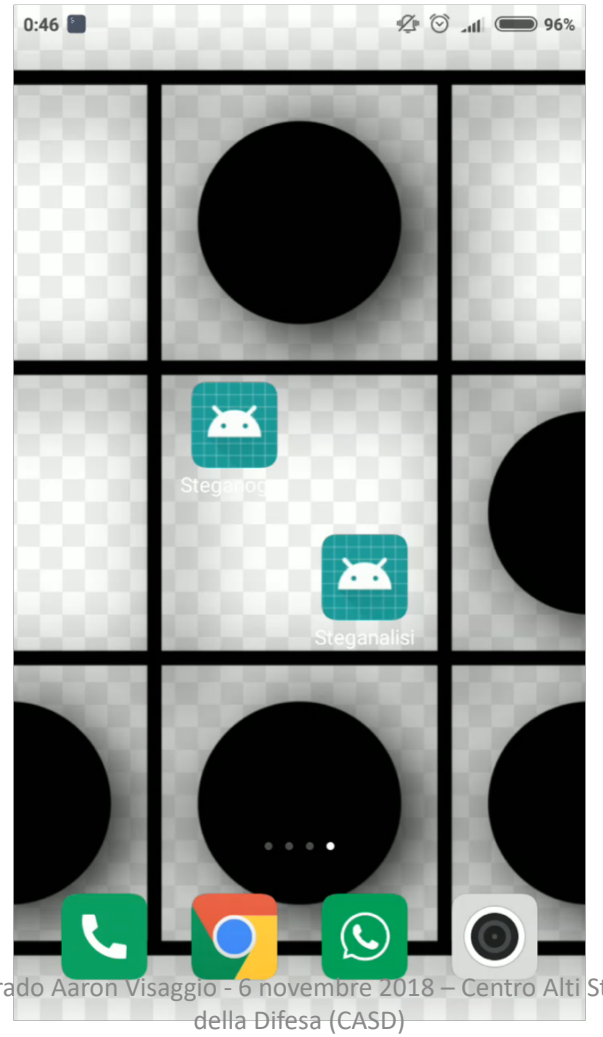
- Blows nuclear power plant (**Stuxnet**)
- Interrupts power supply (**BlackEnergy**)
- Spreads in 15 minutes (**Warhol worm**)
- Commands millions of devices for turning Internet off (**Mirai**)
- Can be purchased as a (modular) Service (**GranCrab**)
- ….


miners
Hypervisor level
Cryptolocker/ransomware
The Malware Universe
fileless
rootkit
spyware
backdoor
implants
Mobile malware
trojan

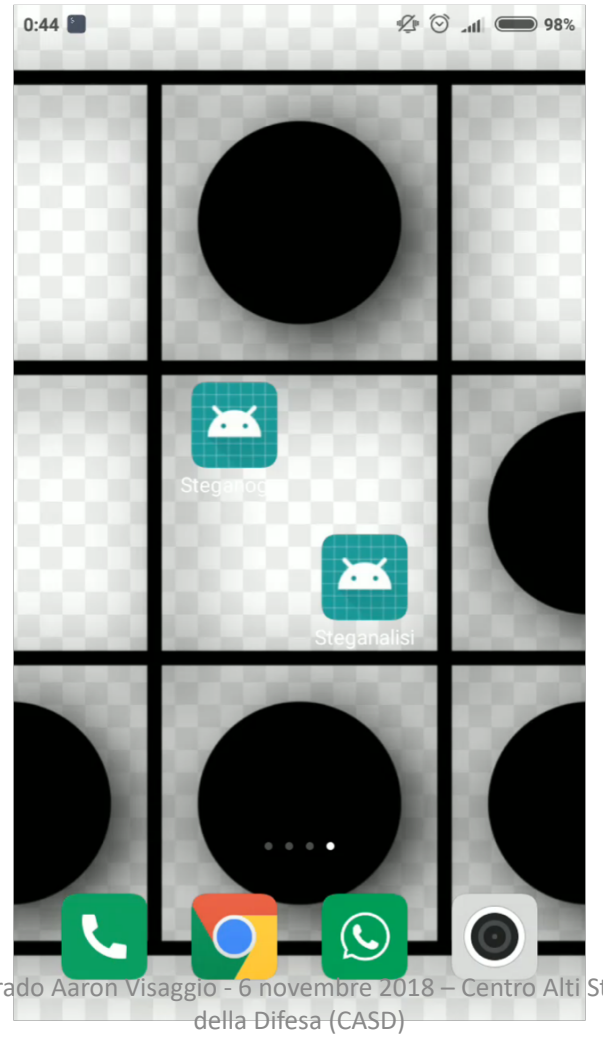
Malicious Software



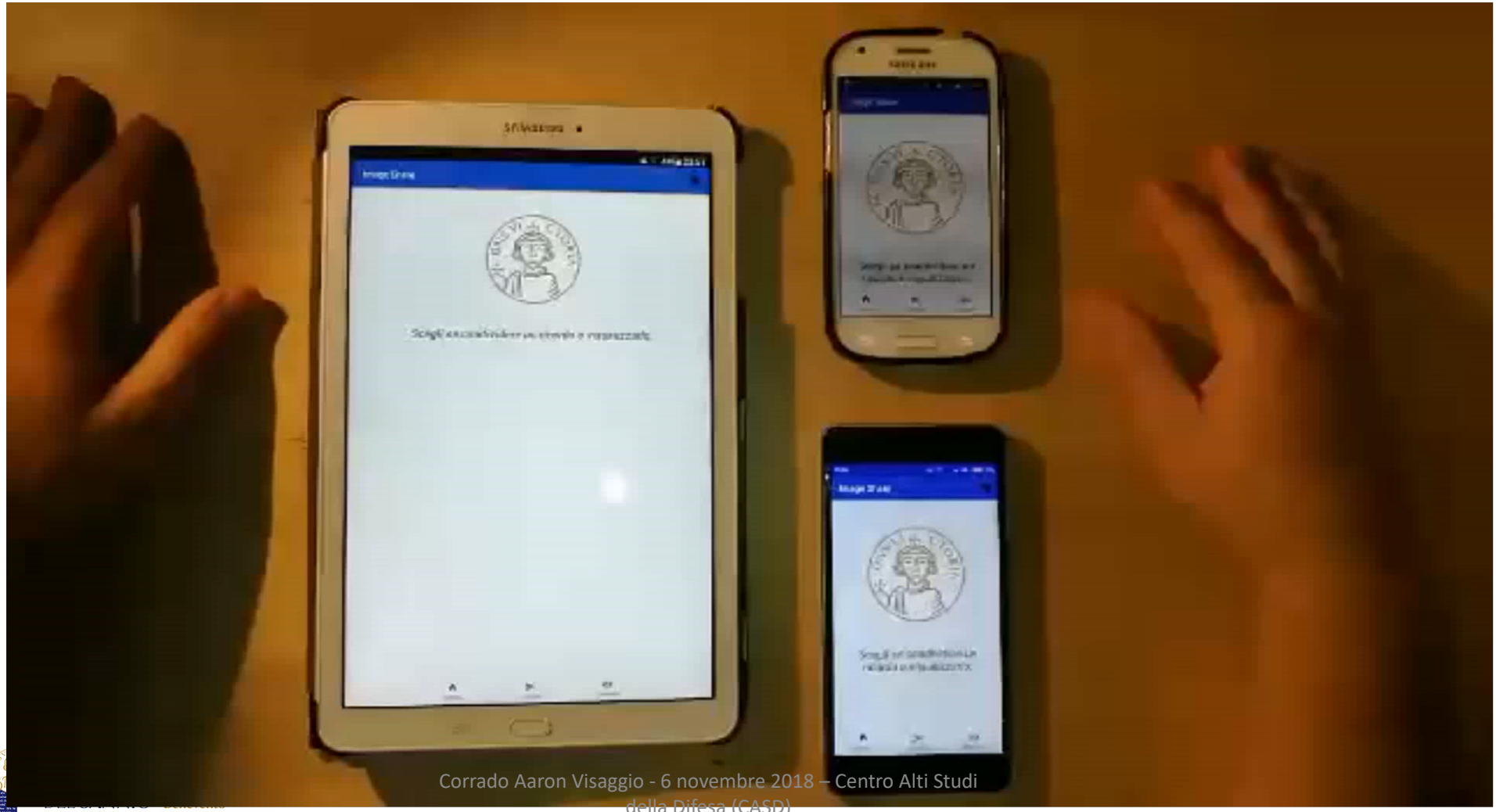
App Steganography



App Steganalysis



Demo



Detection (?)

virustotal

SHA256: 1d42bd83516d1c24d1f24b0889d716f6eb2643e9e8167c4aaaa1de8a4b8b4dd0

Nome del file: app-debug.apk

Rapporto rilevamento: 1 / 58

Data analisi: 2018-10-02 14:44:54 UTC (2 minuti fa)

Analisi | File detail | Ulteriori informazioni | Commenti | Voti

Antivirus	Risultato	Aggiornamento
Babable	PUP.HighConfidence	20180918
Ad-Aware	✓	20181002
AegisLab	✓	20181002
AhnLab-V3	✓	20181002
Alibaba	✓	20180921

Detection (?)

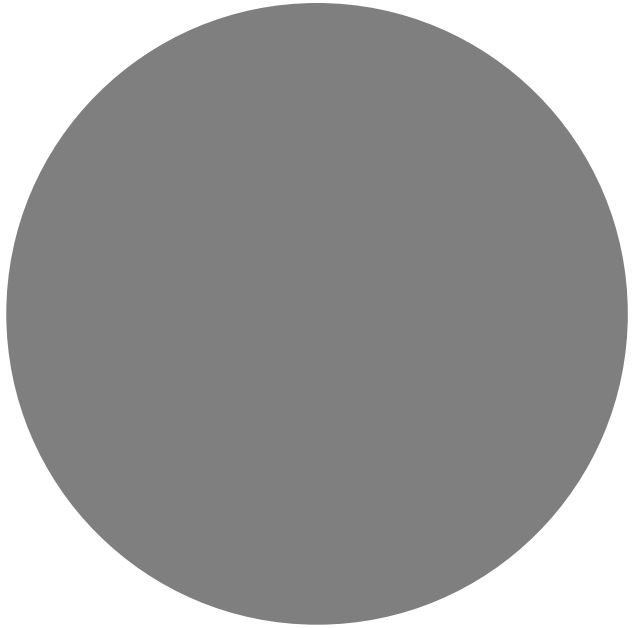
virustotal

SHA256:	c5795fa2b5bcc74422836ba35bee7668a319b87fdf77c1eec4546d90197de89f
Nome del file:	Steg.png
Rapporto rilevamento:	0 / 58
Data analisi:	2018-10-02 15:13:03 UTC (0 minuti fa)

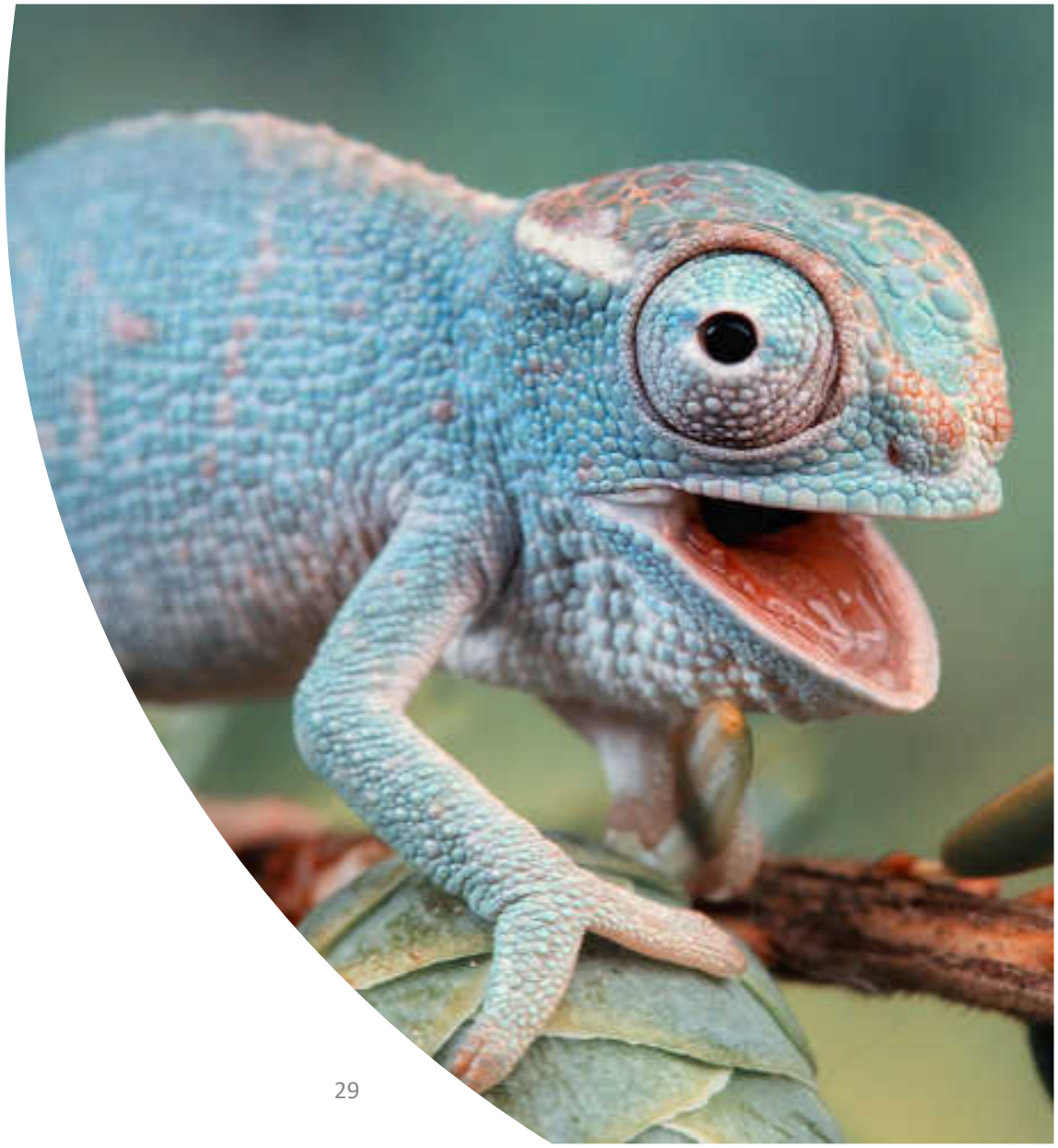
Analisi | Ulteriori informazioni | Commenti | Voti

Antivirus	Risultato	Aggiornamento
Ad-Aware	✓	20181002
AegisLab	✓	20181002
AhnLab-V3	✓	20181002
Alibaba		20180921
ALYac	✓	20181002

Malware Analysis



Malware Analysis main goals



- **Malware Detection**: deciding whether a given sample is malicious
- **Malware Similarity**: to understand how novel samples differ from previous, known ones.
  - Variants detection
  - Family detection
  - Similarities detection
  - Differences detection
- **Malware category detection**: which class of malware the sample belongs to
- **Malware facts**:
  - Origin
  - Author
  - operation

Malware Analysis con Machine Learning

- ***Classification***: it includes two stages, the **model construction** e the **model usage**. The classifier labels the testing set relying on the ***model*** and the extracted ***features***.
- ***Clustering****: to group all the malware that shows* **similar behaviors**. It helps to define the signatures.

Malware Detection…

Pros	Cons
Easy to run Fast identification Broadly accessible Finding comprehensive malware information Hexaustive Not harmful	Failing to detect the polymorphic/encrypted / obfuscated/packed malwares Replicating information in the huge database time window between a malware's release and its detection by anti-malware software tools is about 54 days [Hu 2011].

- ***Signature Based***: Signature-based method identifies unique strings from the binary code [Moskovitch et al. 2009]

… Malware Detection

Pros	Cons
Detecting unconceived types of malware attacks Data-flow dependency detector Detecting the polymorphic malwares	Storage complexity for behavioral patterns Time complexity Coverage limitations Anti-debugging/virtualization techniques

- ***Behavior Based***: catch execution patterns and characteristics by exploding malware in virtualized environments imitating conditions of susceptible systems to infection

General Information

Joe Sandbox Version:	26.0.0 Aquamarine
Analysis ID:	961422
Start date:	20.09.2019
Start time:	13:28:47
Joe Sandbox Product:	Cloud
Overall analysis duration:	0h 9m 19s
Hypervisor based Inspection enabled:	false
Report type:	full
Sample file name:	Nuovo_documento_2019.09.20.doc
Cookbook file name:	defaultwindowsofficecookbook.jbs
Analysis system description:	Windows 7 (**Office 2010 SP2**, Java 1.8.0_40 1.8.0_191, Flash 16.0.0.305, Acrobat Reader 11.0.08, Internet Explorer 11, Chrome 55, Firefox 43)
Number of analysed new started processes analysed:	13
Number of new started drivers analysed:	0
Number of existing processes analysed:	0
Number of existing drivers analysed:	0
Number of injected processes analysed:	0
Technologies:	• HCA enabled • EGA enabled • HDC enabled • GSI enabled (VBA) • AMSI enabled
Analysis stop reason:	Timeout
Detection:	MAL
Classification:	mal100.bank.evad.winDOC@18/52@1/3
EGA Information:	• Successful, ratio: 100%
HDC Information:	• Successful, ratio: 90.5% (good quality ratio 87.9%) • Quality average: 82.2% • Quality standard deviation: 25.6%
HCA Information:	• Successful, ratio: 84% • Number of executed functions: 133 • Number of non-executed functions: 301
Cookbook Comments:	• Adjust boot time • Enable AMSI • Found application associated with file extension: .doc • Found Word or Excel or PowerPoint or XPS Viewer • Attach to Office via COM • Scroll down • Close Viewer
Warnings:	Show All

Antivirus or Machine Learning detection for sample	Hide sources
Source: Nuovo_documento_2019.09.20.doc	Joe Sandbox ML: detected

Multi AV Scanner detection for dropped file		Hide sources
Source: C:\Users\user\982.exe	Virustotal: Detection: 15%	Perma Link

Multi AV Scanner detection for submitted file		Hide sources
Source: Nuovo_documento_2019.09.20.doc	Virustotal: Detection: 22%	Perma Link

Cryptography:

Uses Microsoft's Enhanced Cryptographic Provider		Hide sources
Source: C:\Users\user\982.exe	Code function: 7_2_0040207B CryptDuplicateHash,CryptEncrypt,CryptDestroyHash,	7_2_0040207B
Source: C:\Users\user\982.exe	Code function: 7_2_00401F56 CryptGetHashParam,	7_2_00401F56
Source: C:\Users\user\982.exe	Code function: 7_2_0040215A CryptDuplicateHash,CryptDecrypt,CryptVerifySignatureW,CryptDestroyHash,	7_2_0040215A
Source: C:\Users\user\982.exe	Code function: 7_2_00401F75 CryptAcquireContextW,CryptImportKey,LocalFree,CryptReleaseContext,	7_2_00401F75
Source: C:\Users\user\982.exe	Code function: 7_2_00401F11 CryptExportKey,	7_2_00401F11
Source: C:\Users\user\982.exe	Code function: 7_2_00401FFC CryptGenKey,CryptCreateHash,CryptDestroyKey,CryptDestroyKey,CryptReleaseContext,	7_2_00401FFC
Source: C:\Windows\System32\sortedwatched.exe	Code function: 12_2_00401F75 CryptAcquireContextW,CryptDecodeObjectEx,CryptImportKey,LocalFree,CryptReleaseContext,	12_2_00401F75
Source: C:\Windows\System32\sortedwatched.exe	Code function: 12_2_00401FFC CryptGenKey,CryptCreateHash,CryptDestroyKey,CryptDestroyKey,CryptReleaseContext,	12_2_00401FFC
Source: C:\Windows\System32\sortedwatched.exe	Code function: 12_2_0040207B CryptDuplicateHash,CryptEncrypt,CryptDestroyHash,	12_2_0040207B
Source: C:\Windows\System32\sortedwatched.exe	Code function: 12_2_00401F56 CryptGetHashParam,	12_2_00401F56
Source: C:\Windows\System32\sortedwatched.exe	Code function: 12_2_0040215A CryptDuplicateHash,CryptDecrypt,CryptVerifySignatureW,CryptDestroyHash,	12_2_0040215A
Source: C:\Windows\System32\sortedwatched.exe	Code function: 12_2_00401F11 CryptExportKey,	12_2_00401F11
Source: C:\Windows\System32\sortedwatched.exe	Code function: 12_1_00401F75 CryptDecodeObjectEx,LocalFree,	12_1_00401F75

Spreading:

Enumerates the file system	Show sources

Software Vulnerabilities:

Potential document exploit detected (performs DNS queries)	Show sources
Potential document exploit detected (performs HTTP gets)	Show sources
Potential document exploit detected (unknown TCP traffic)	Show sources

Potential document exploit detected (unknown TCP traffic)	Show sources

Networking:

Detected TCP or UDP traffic on non-standard ports	Hide sources
Source: global traffic	TCP traffic: 192.168.1.16:49164 -> 149.167.86.174:990

IP address seen in connection with other malware	Hide sources
Source: Joe Sandbox View	IP Address: 149.167.86.174 149.167.86.174

JA3 SSL client fingerprint seen in connection with other malware	Hide sources
Source: Joe Sandbox View	JA3 fingerprint: 05af1f5ca1b87cc9cc9b25185115607d

Connects to IPs without corresponding DNS lookups	Show sources
Contains functionality to download additional files from the internet	Show sources
Downloads files	Show sources
Performs DNS lookups	Show sources
Urls found in memory or binary data	Show sources
Uses HTTPS	Show sources

Key, Mouse, Clipboard, Microphone and Screen Capturing:

Contains functionality to retrieve information about pressed keystrokes	Show sources

E-Banking Fraud:

Detected Emotet e-Banking trojan	Show sources

Spam, unwanted Advertisements and Ransom Demands:

Contains functionality to import cryptographic keys (often used in ransomware)	Show sources

System Summary:

Office document tries to convince victim to disable security protection (e.g. to enable ActiveX or Macros)	Show sources
Document contains an embedded VBA macro which may check the recent opened files (possible anti-VM)	Show sources
Powershell drops PE file	Show sources

Created / dropped Files

C:\ProgramData\Microsoft\Crypto\RSA\S-1-5-18\6d14e4b1d8ca773bab785d1be032546e_041d84af-7e76-450d-8340-55db3c73c359	
Process:	C:\Windows\System32\sortedwatched.exe
File Type:	data
Size (bytes):	2134
Entropy (8bit):	7.082471693816772
Encrypted:	false
MD5:	7B2759997F3D8E28C124D04DC495C0B5
SHA1:	30F9D822FC7B2A2E6A2EC1767949F739BF9CBC4C
SHA-256:	464553C0BA166E1C354DDC6477C6D466584F37E3367442B4653ACFA5D7234B7D
SHA-512:	57858DA819CB6F76C9F5CCAECDC06784F04960E39E78FEBA6E9CECBCC644B8242CE1DD90AB8919133DD789DFDF54A546CAE72EFA54F9E590ACBB013800CDCC43
Malicious:	false
Reputation:	low
Preview:	.....\.....SYSTEM.....RSA1H.....?.....}...h8...B~k..!.R..<.HN:D...tW....5g.n.xLu5..tl. .q5e.. .....z..O.....E...g@....V..$.....,...C.r.y.p.t.o.A.P.I. .P.r.i.v.a.t.e. .K.e.y....f..... .....6..h.N..Z....kN...G..$+..b..... ...r.....^...'-RoZM.#.^.S.....O.W..$(.0.4/.v{$.{;..n.....T..[.,....)...cB./(2.~...H|.7..f..k ...@...I..D?..*....zNJi.....g}d...6.h.QkW.Q.X...6...{}*.[*G...0.|.\.7..i{.....p...``u.....$...B.c.h.....N.....n...p.. .....D.p.?....NR.Vo|.ef....x&.D. .`.....B.s].{....:(.. ....(&..c9.*x. ).s.~.D.S..C.^L....{...PHP...#..N..L.[.u.?N.....v.....M .S;0f...JZ6Wf.....P..)*.#....d`.=.Q..5Y&.o...@....h:...,S+lGg4..p27..*... !.Ci.W.s.J.ulu.$`2Vk.).....{,.....z.. O.....E...g@....V..$.....E.x.p.o.r.t. .F.l.a.g.....f..... ....|.Q.o..,Y..T.7W..|..p.u....$x..... ..../.....}..+..C?.j.!vg..O..U....O.....Q-....&@

C:\Users\user\982.exe	
Process:	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
File Type:	PE32 executable (GUI) Intel 80386, for MS Windows
Size (bytes):	425472
Entropy (8bit):	6.712476322966454
Encrypted:	false
MD5:	3A74A93E7831D0953B5CEFB9C98505F1
SHA1:	C74D84DE41D9294DA948D3CAACDDED254853E57C
SHA-256:	8743FB2C992EE623779B119C5BB06F9A523E2F335B0E64B8E133C4867295CE3C
SHA-512:	DA385FEAC0E13C7D8F4A7BECC92EDA980D160E0FF570F6193E111D3D5EB14B423CBC8329C146ECA01D27251B83DEB8E3ACE00FD3008935420B5767F1EE195290
Malicious:	**true**
Antivirus:	• Antivirus: Virustotal, Detection: 16%, Browse
Joe Sandbox View:	• Filename: DZB_V176H033B3E4VU_LN.doc, Detection: malicious, Browse
Reputation:	low
Preview:	MZ.....@.....!..L.!This program cannot be run in DOS mode....$.....Z...Z...Z...Z...Z...Z...Z...Z...Z...Z...Z..HZ...Z..^Zu..Z..YZm..Z..IZ...Z..LZ...ZRich...Z.....PE..L... k.].....z.....@.....@..... ..z..... ..05..@.....x...@.....text..... ..`.rdata.....@..@.data...Xj.....(.....@....rsrc...z.... .....@..@.....

C:\Users\user\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.MSO\13E77F69.wmf	
Process:	C:\Program Files\Microsoft Office\Office14\WINWORD.EXE
File Type:	Targa image data - Map - RLE 28 x 65536 x 0 +5 "\004"
Size (bytes):	444
Entropy (8bit):	3.286841866831989

Public

IP	Country	Flag	ASN	ASN Name	Malicious
149.167.86.174	Australia		45510	unknown	false
198.49.65.242	United States		33182	unknown	false
181.164.8.25	Argentina		10318	unknown	false

Static File Info

General

File type:	Composite Document File V2 Document, Little Endian, Os: Windows, Version 6.1, Code page: 1252, Author: Joseph Fritsch, Template: Normal.dotm, Revision Number: 1, Name of Creating Application: Microsoft Office Word, Create Time/Date: Fri Sep 20 08:30:00 2019, Last Saved Time/Date: Fri Sep 20 08:30:00 2019, Number of Pages: 1, Number of Words: 95, Number of Characters: 547, Security: 0
Entropy (8bit):	6.727007621624430 6
TrID:	• Microsoft Word document (32009/1) 52.89% • Microsoft Word document (old ver.) (19008/1) 31.41% • Generic OLE2 / Multistream Compound File (8008/1) 13.23% • Java Script embedded in Visual Basic Script (1500/0) 2.48%
File name:	Nuovo_documento_2019.09.20.doc
File size:	236544
MD5:	1b9714114ff735277c8981c84d4f2393
SHA1:	beacaf09fb062e5f3e986ee294ed5ec97fc26c12
SHA256:	beb82d8b2429911fffe39457bd4bb8bbe033ca34826df10b291fa74b33c7275a
SHA512:	49f4a4e8de51483e8f5500e42be792bec9e26d5e583d88011d9dc732a5a504adb4ac34cccf2d3382b802d871acf038063cc850031c2ae910b55413d2a2358070
SSDEEP:	6144:+d96T4Rci2R9JtXvlj++PWVI1dGLkIV7NSU4jJntATfDDBpp:+d96T4Rci2R9JtXvH+PWVI1SXV7NSU4+
File Content Preview:	.....> ...

File Icon

Icon Hash:	e4eea2aaa4b4b4a4

Static OLE Info

General

Document Type:	OLE

File Created

File Deleted

File Written

File Read

File Attributes Queried

Directory Queried

Other File Operations

Volume Information Queried

Section Activities — Show windows behavior

Sections loaded by Windows

Sections loaded by Program

Registry Activities — Show windows behavior

Key Opened

Key Created

Key Deleted

Key Value Created

Key Value Modified

Key Value Deleted

Tool Chain for Malware Analysis

- File Static Analyzer
- Network Packet inspection
- Memory Forensics
- Registry utility
- Process Analyzer
- Malicious web page explorer
- Sandboxes

Threat Intelligence

- “Threat intelligence is the output of analysis based on identification, collection, and enrichment of relevant data and information regarding cyber attacks.”
- Threat intelligence falls into two categories.
  - Operational intelligence is produced by computers
  - Strategic intelligence is produced by human analysts.
- The two types of threat intelligence are heavily interdependent

Threat Intelligence

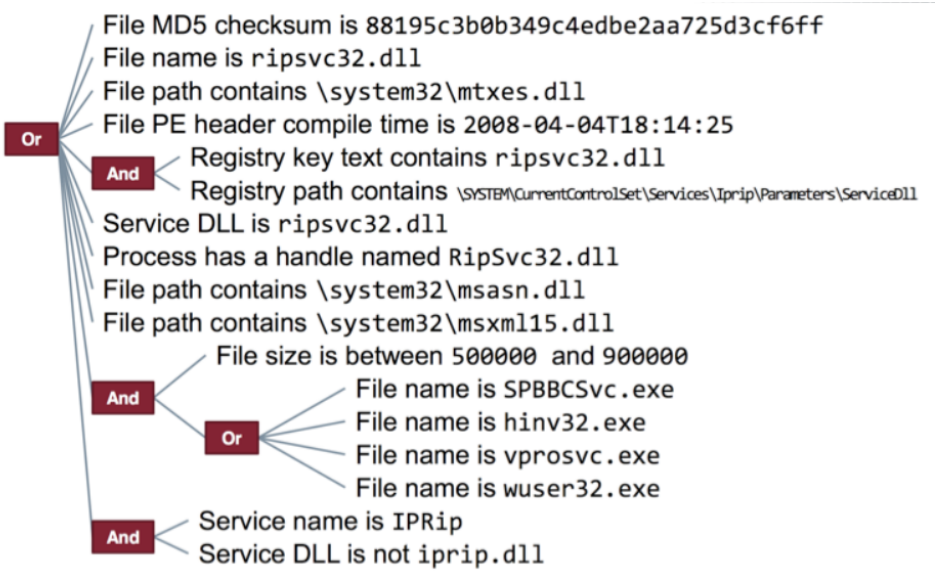
- Operational Intelligence:
  - A common example of operational threat intelligence is the automatic detection of distributed denial of service (DDoS) attacks, whereby a comparison between indicators of compromise (IOCs) and network telemetry is used to identify attacks much more quickly than a human analyst could.
- Strategic Intelligence:
  - focuses on the much more difficult and cumbersome process of identifying and analyzing threats to an organization's core assets, including employees, customers, infrastructure, applications, and vendors.

Threat Intelligence

- What share?
  - Indicators of Compromise – IoC
    - «is an artifact observed in a system or on a network that with high confidence indicates a compromission»
    - IP, URLs, Virus signatures, Hashes, Malware files
  - Tactics, Techniques and Procedures – TTP
    - «Are representations of the behavior or modus operandi of cyber adversaries»
  - Report
  - IDPS rules

Threat Intelligence

- How share?
  - OpenIoc
  - Yara
  - TAXII
  - Stix
  - Cybox



Source:
https://threatpost.com/misunderstanding-indicators-of-compromise/117560/

Threat Intelligence

- Yara rules example

```
import "pe"

rule BadRabbit_dropper {

    meta:
        description = "Yara Rule for Bad Rabbit dropper identification"
        author = "CSE CybSec Enterprise - Z-Lab"
        last_updated = "2017-10-31"
        tlp = "white"
        category = "informational"

    strings:
        // Flash string
        $flash = "Flash" wide

        // File infpub extracted
        $a = "C:\\Windows\\infpub.dat" wide
        $b = "infpub.dat" wide

        // Execution of infpub.dat
        $c = "%ws C:\\Windows\\%ws,#1 %ws" wide

    condition:
        all of them and
        pe.version_info["ProductName"] contains "Installer/Uninstaller"
}
```

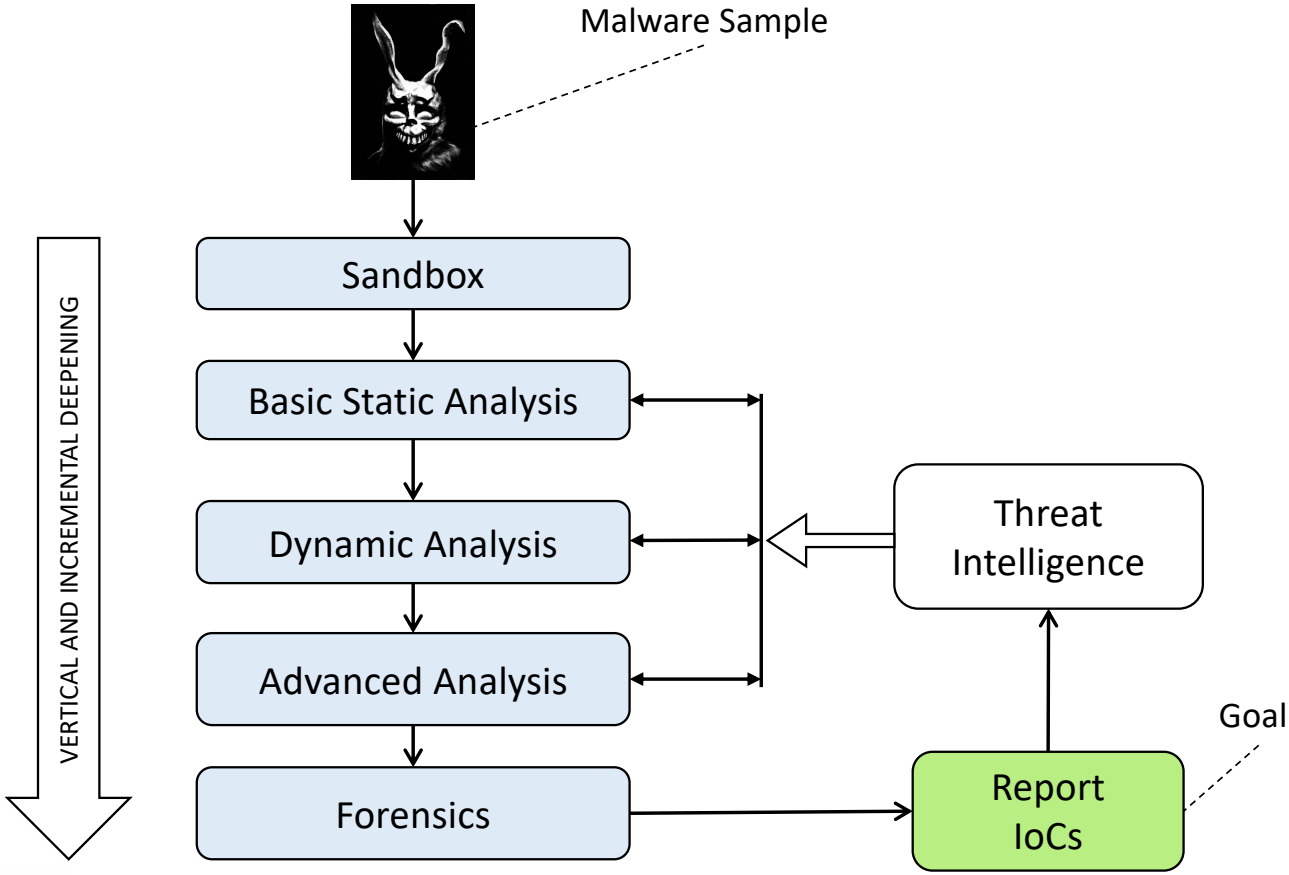
Threat Intelligence Platforms

- Where share?
  - OTX by Alienvault
  - XForce by IBM
  - MISP
  - Anomali
  - Threatcrowd
  - Threatconnect
  - Blueliv

Threat Intelligence & IoC

- How extract malwares' IoCs and other characteristics to share?

The Malware Analysis Process



Malware Sample

Retrieve malware from:

- Infected machines
- Disk images
- Network traffic
- Suspicious files
- Public sources
- Deep web
- HoneyNet

Sandbox

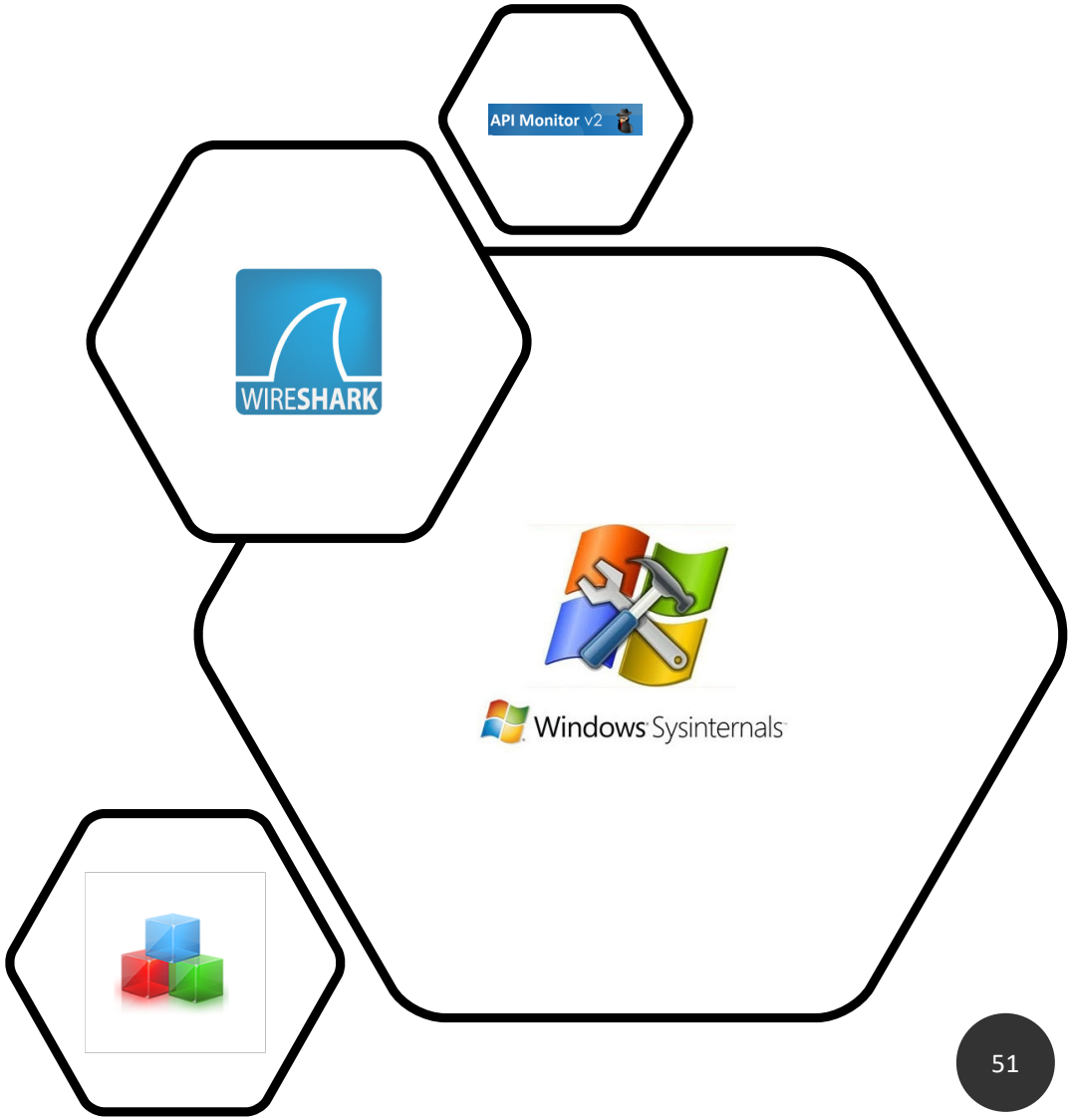
- Submit the sample into Cuckoo Sandbox private instance or Payload Security
- Malware's first impressions and initial triaging

Basic Static Analysis

- Retrieve the first info's about the characteristics of the malicious file:
  - FileType
  - Hashes
  - Strings
  - Sections
  - Imports
  - Packers

Dynamic Analysis

- Observe the malware in action:
  - Runtime API calls
  - Network Traffic
  - Files' accesses
  - Registry Keys' accesses
  - System settings alteration
  - Disk Modification
  - Lateral movements
  - Privilege Escalation



Advanced Analysis

- Advanced Static&Behavioural Analysis
  - Refine the characteristics of the malware through the correspondence of the malware execution in a debugger with its disassembled code
  - Find particular structures and IoC in the malware's code

Forensics

- Extract evidences and digital artifacts from various supports
  - Disk
  - Memory
  - Volatility Framework

IoC extraction

- Synthetize the info about malwares to recognize them in rules that allow their detection
  - Yara
  - OpenIoC

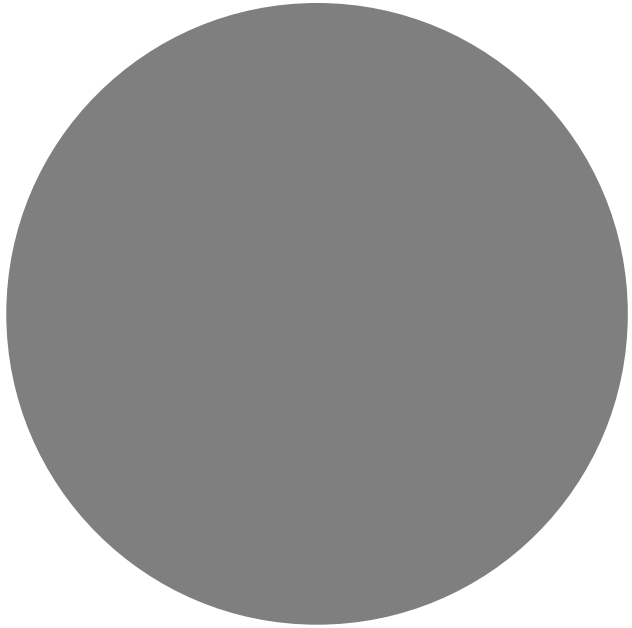
Gather intelligence from reports and IoCs

- Take comparison in all previous steps between the info gathered in various Threat Intelligence Platforms and those extracted during the analysis
  - Compare hashes
  - Compare behaviour
  - Compare IoCs

Features

- **Windows API & System Calls:** Windows API calls are used by almost all programs to send the requests to the operating system -> can reflect the behavior of the program
- **N-grams:** N-grams are all substrings in the program code with a length of *N*
- **Strings:** The interpretable strings are the high-level specifications of malicious behaviors. These strings can reflect the attacker's intent and goal since they often contain the important semantic information
- **Opcodes:** An OpCode (i.e., Operational Code) is the subdivision of a machine language instruction that identifies the operation to be executed
- **Control Flow Graphs (CFGs):** A CFG is a graph that represents the control flow of a program.
- **Network Activity**: used protocols, TCP/UDP ports, HTTP requests, DNS-level in- teractions
- **File System**: how many files are read or modified, what types of files and in what directories, and which files appear in not-infected/infected machines
- **CPU registers**: whether any hidden register is used, and what values are stored in the registers, especially in the FLAGS register
- **PE file characteristics*:*** sections, imports, symbols, used compilers

Machine Learning Algorithms



Main Concepts…

- ML algorithms are generally used to ***identify patterns across data***.
- Data is usually represented as a list of examples:

  $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$
- Here, $x_1, \ldots, x_n$ are n observations and $y_1, \ldots, y_n$ are the responses to those observations and x_i is a vector of size p.
- x_1 can represent some measurements of a given binary while y_1 can represent the label *malicious* or *benign*.

$$X = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{12} & \ldots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{pmatrix} \qquad Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

… Main Concepts…

- We assume that there is a relation between the data defined as:

$$Y = g(X) + \varepsilon$$

- Where ε represents an error term. Therefore, the goal of machine learning is to find:

$$\hat{Y} = \hat{g}(X)$$

- Where $\hat{g}$ is the best estimation for g, and $\hat{Y}$ are the estimations for the true Y .
- Machine learning algorithms are usually used to make ***future predictions***. These algorithms work by building a prediction model and test it
- The dataset available during the construction of one prediction model is usually divided into 3 sets with different purposes: ***training set***, ***validation set*** and ***testing set***.
- The usual separation of the available dataset is: 60% for the training set, 20% for the validation set and 20% for the testing set.

… Main Concepts

- One important thing to note is that with more training and validation data, the variance of the generated model will be reduced while increasing the size of the test set will decrease the variance on the results.
- **Training Set.** The training set is the set of examples used to fit the model chosen to be the prediction model.
- **Validation Set.** The validation set is the set of examples used to predict the responses with the model created with the training set.
- **Testing Set.** The testing set is a set of examples used to report the performance obtained once the ML algorithm is optimised and ready to be deployed.

- ***Overfitting*** is a problem caused by ML algorithms that learn the patterns on the training data too precisely and ***are not able to generalise on new and unseen instances***.
- Overfitting causes the error obtained on the test data to increase, instead of decreasing, when more training data is added to the training set.

Cross Validation

- ***Cross-validation (CV)*** is a re-sampling technique that is used for model validation. It is used to ***estimate the test error*** of a model. It involves *holding out* a subset of training samples from the training process and then testing the model created with this subset.
- ***K-fold cross validation*** consists of dividing the dataset in *k* non-overlapping folds. The model will be trained with the *k* − 1 folds and the remaining one will be used to test the model.

Malware as a Commodity

Underground marketplace price list

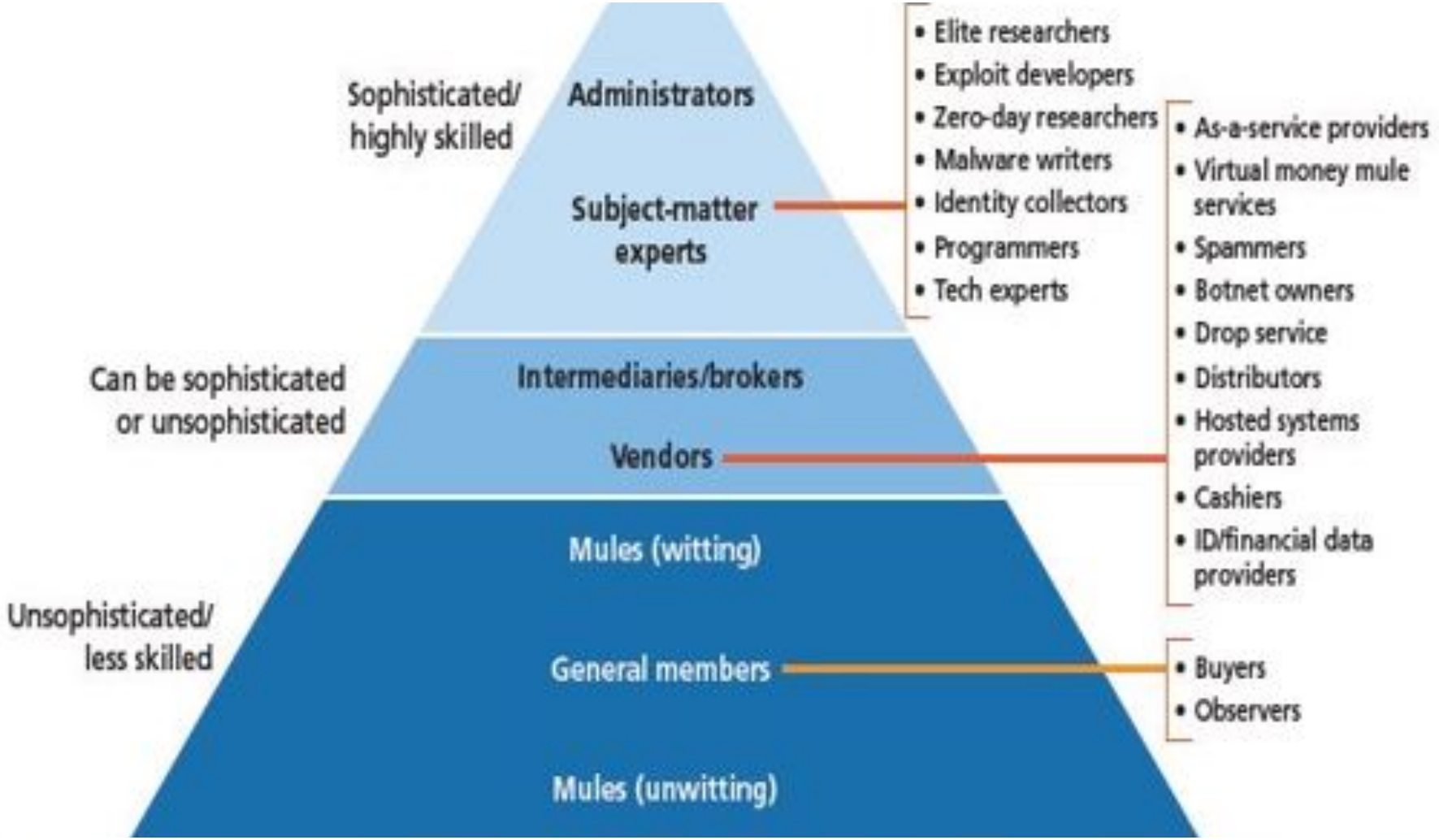
Payment cards	Price
Single credit card	$0.5 - $30
Single credit card with full details (Fullz)	$20 - $60
Dump of magnetic strip track 1&2 & PIN	$60 - $100
Malware	
Basic banking Trojan kit with support	$100
Password stealing Trojan	$25 - $100
Android banking Trojan	$200
Office macro downloader generator	$5
Malware crypter service (make hard to detect)	$20 - $40
Ransomware kit	$10 - $1800
Services	
Media streaming services	$0.10 - $10
Hotel reward program accounts (100K points)	$10 - $20
Airline frequent flyer miles account (10K miles)	$5 - $35
Taxi app accounts with credit	$0.5 - $1
Online retail gift cards	20% - 65% of face value
Restaurant gift cards	20% - 40% of face value
Airline ticket and hotel bookings	10% of face value
DDoS service, < 1hr duration, medium target	$5 - $20
DDoS service, > 24hr duration, medium & strong target	$10 - $1000
Dedicated bulletproof hosting (per month)	$100 - $200

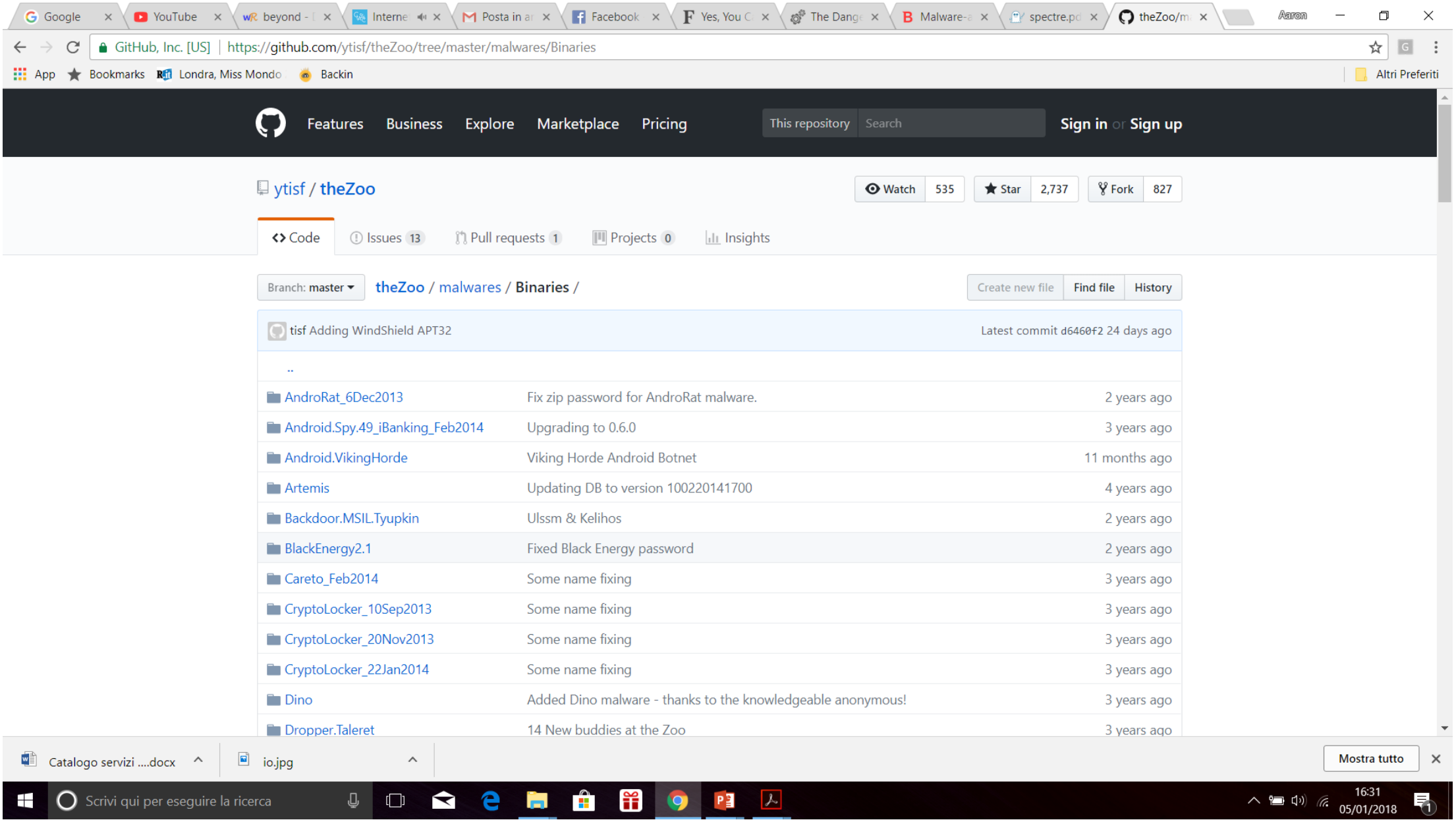
Money transfer services	
Cash-out service	10% - 20%
Accounts	
Online bank accounts	0.5% - 10% of account balance
Retailer accounts	$20 - $50
Cloud service provider accounts	$6 - $10
Identities	
Identity (Name, SSN & DOB)	$0.1 - $1.5
Scanned passports and other documents (e.g. utility bill)	$1 - $3

Malware as a Service (MAAS)

- cybercrooks can expect to **earn 1,425% return on investment** from a 30-day malware infection campaign
- Malware-as-a-service (MaaS) is the business delivery model that today's online black market relies heavily upon.
- Using the MaaS model, anyone who is willing to commit a crime on line is now able to launch attacks on internet users and corporations anywhere in the world. And they **don't even need to have the necessary equipment or skills to do that**.

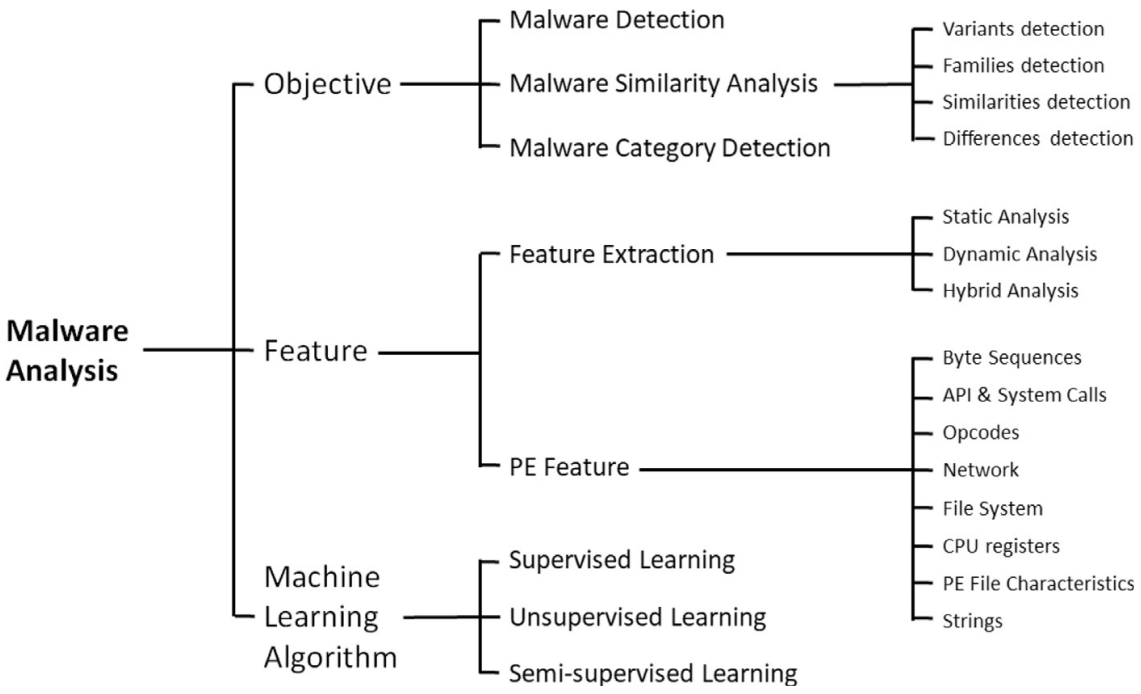
Source: https://blog.finjan.com/the-dangers-of-maas-malware-as-a-service/


Sophisticated/ highly skilled
Administrators
Subject-matter experts
Can be sophisticated or unsophisticated
Intermediaries/brokers
Vendors
Unsophisticated/ less skilled
Mules (witting)
General members
Mules (unwitting)
• Elite researchers
• Exploit developers
• Zero-day researchers
• Malware writers
• Identity collectors
• Programmers
• Tech experts
• As-a-service providers
• Virtual money mule services
• Spammers
• Botnet owners
• Drop service
• Distributors
• Hosted systems providers
• Cashiers
• ID/financial data providers
• Buyers
• Observers


Google
YouTube
beyond -
Interne
Posta in ar
Facebook
Yes, You C
The Dange
Malware-a
spectre.pd
theZoo/m
Aaron
GitHub, Inc. [US] | https://github.com/ytisf/theZoo/tree/master/malwares/Binaries
App
Bookmarks
Londra, Miss Mondo
Backin
Altri Preferiti
Features
Business
Explore
Marketplace
Pricing
This repository
Search
Sign in or Sign up
ytisf / theZoo
Watch 535
Star 2,737
Fork 827
Code
Issues 13
Pull requests 1
Projects 0
Insights
Branch: master
theZoo / malwares / Binaries /
Create new file
Find file
History
tisf Adding WindShield APT32
Latest commit d6460f2 24 days ago
..
AndroRat_6Dec2013 Fix zip password for AndroRat malware. 2 years ago
Android.Spy.49_iBanking_Feb2014 Upgrading to 0.6.0 3 years ago
Android.VikingHorde Viking Horde Android Botnet 11 months ago
Artemis Updating DB to version 100220141700 4 years ago
Backdoor.MSIL.Tyupkin Ulssm & Kelihos 2 years ago
BlackEnergy2.1 Fixed Black Energy password 2 years ago
Careto_Feb2014 Some name fixing 3 years ago
CryptoLocker_10Sep2013 Some name fixing 3 years ago
CryptoLocker_20Nov2013 Some name fixing 3 years ago
CryptoLocker_22Jan2014 Some name fixing 3 years ago
Dino Added Dino malware - thanks to the knowledgeable anonymous! 3 years ago
Dropper.Taleret 14 New buddies at the Zoo 3 years ago
Catalogo servizi ....docx
io.jpg
Mostra tutto
Scrivi qui per eseguire la ricerca
16:31
05/01/2018

Malware Analysis & Machine Learning

- From «Ucci, D., Aniello, L., & Baldoni, R. (2018). Survey of machine learning techniques for malware analysis. *Computers & Security*»



Dealing with the Unknown

- ***Concept drift:*** ***new variants*** *of a known malware deteriorate the performances of a classifier over time****.***
  - ***A training set may include out-of-date instances***
  - ***A training set may include an insufficient number of instances***
- ***Unknown***: malware **not used** for **training** the classifier
- **Research Question**: which is the ***resilience*** of a classifier when faces an **unknown** malware?
- **Dataset**: Microsoft Kaggle[1] database (10869 instances belonging to 9 families)
- **Process of Analysis**: PCA + Binary Classification with Unknown + Binary Classificatoin with Known.

[1]https://github.com/albertsanso/kaggle-microsoft-malware/tree/master/malware/2nditeration

Resilience

Accuracy a = $\frac{TP+TN}{TP+FP+TN+FN}$

Resilience R= $a_{w/o\_unknown}$-$a_{w\_unknown}$

Ramnit							
Model	Lollipop	Kelihos v. 3	Vundo	Tracur	Kelihos v.1	Obfuscator	Gatak
KNeighbors	23.73	77.45	16.6	9.07	6.28	18.05	39.97
Perceptron	9.77	0.069	26.25	18.84	27.33	17.15	17.55
Stochastic gradient descent	13.12	14.74	8.93	14.71	24.82	14.77	3.42
Gaussian Naive Bayes	8.6	0.06	33.26	24.89	0.00	16.43	0.83
logistic regression	11.65	1.32	8.45	11	30.6	9.98	0.18
support vector machine	0.39	2.00	0.03	18.30	1.01	5.03	0.86
decision tree	25.70	53.32	23.78	17.33	82.86	4.74	8.94
random forest	26.08	81.33	7.48	9.39	1.01	4.31	11.94
Lollipop							
Model	Ramnit	Kelihos v.3	Vundo	Tracur	Kelihos v.1	Obfuscator	Gatak
KNeighbors	32.84	24.52	25.58	28.91	2.26	24.93	0.26
perceptron	5.69	9.77	18.35	18.94	1.3	14.19	21.61
Stochastic gradient descent	4.97	0.23	16.42	17.4	1.3	8.71	33.97
Gaussian Naive Bayes	8.46	0.02	9.68	5.64	7.48	2.29	0.68
logistic regression	10.38	4.23	15.45	7.87	3.46	14.1	21.59
support vector machine	6.2	0.11	9.55	2.26	1.3	4.69	0.19
decision tree	32.8	18.12	5.18	18.49	18.08	15.25	19.03
random forest	13.89	2.09	12.43	18.66	19.34	8.82	1.28
Kelihos v. 3							
Model	Ramnit	Lollipop	Vundo	Tracur	Kelihos v.1	Obfuscator	Gatak
KNeighbors	5.36	47.05	0.21	1.73	-	6.55	12.24
perceptron	37.81	60.79	34.81	30.55	-	5.53	56.15
Stochastic gradient descent	27.98	50.29	28.44	15.05	-	6.98	77.87
Gaussian Naive Bayes	0.88	0.70	0.17	1.52	-	0.030	2.99
logistic regression	23.67	54.55	30.41	28.09	-	5.90	50.64
support vector machine	6.99	29.44	8.74	2.02	-	5.28	39.32
decision tree	12.05	3.15	4.63	3.33	-	0.140	4.82
random forest	1.08	16.78	0	0.53	-	0.32	0
Vundo							
Model	Ramnit	Lollipop	Kelihos v.3	Tracur	Kelihos v.1	Obfuscator	Gatak
KNeighbors	0.32	39.44	10.19	23.97	31.68	23.04	30.84
perceptron	3.26	61.14	13.33	55.27	16.68	13.64	0.29
Stochastic gradient descent	14.96	50.44	4.89	44.59	82.73	7.8	0.19
Gaussian Naive Bayes	88.95	86.12	99.9	85.26	69.35	67.63	96.69
logistic regression	11.1	62.96	3.98	48.59	24.37	20.94	1.82

Machine learning is strongly sensitive to the unknown.

Concluding Remarks

2 biases in malware classification

- ***Spatial bias*** refers to unrealistic assumptions about the ratio of goodware to malware in the data.
- ***Temporal bias*** refers to temporally inconsistent evaluations which integrate future knowledge about the testing objects into the training phase or create unrealistic settings.
- The ***base-rate fallacy*** [Axelsson, 2000] describes how evaluation metrics such as TPR and FPR are misleading in intrusion detection, due to significant class imbalance (most traffic is benign).
- Pendlebury and colleagues [Pendlebury et al., 2019]experimentally verify on a dataset of 129K apps (with 10% malware) that, due to bias, performance can decrease up to 50% in practice in two well-known Android malware classifiers, DREBIN [Arp et al., 2014] and MAMADROID [Mariconti et al., 2017]

Main issues regarding dataset

- Quick **obsolescence** of dataset
- **Incomplete/ imbalanced** coverage of dataset
  - Windows vs Linux/MacOS
  - Android vs IOS
  - Workstation vs SCADA
- **Representativeness** of population
  - IOT is a typical example
- To which extent can we **trust** the dataset?
- How much does the dataset **polarize** the research strategy and goals?

Challenges and future directions

- ***Incremental learning*****:** how to update timely and continuously the training set and how this affect the classifiers
- ***Active learning***: In malware detection, there is limited research using active learning to select a representative sample(s) from the large file sample collection.
- ***Prediction of malware prevalence***: how to forecast malware trends
- ***Adversarial learning*****:** how to develop techniques that are robust and secure in adversarial scenarios
- ***Malware Attribution***: characterizing the authorship of a malware
- ***Malware Triage***: methods to identify prioritization logics in malware analysis
- ***Malware detection in its infancy***: infamous campaign were realized with malware submitted to publicly available sandboxes months or years before.
- ***Finding new features:*** identifying a smaller set of features with greater efficacy for classification and detection

Bibliography…

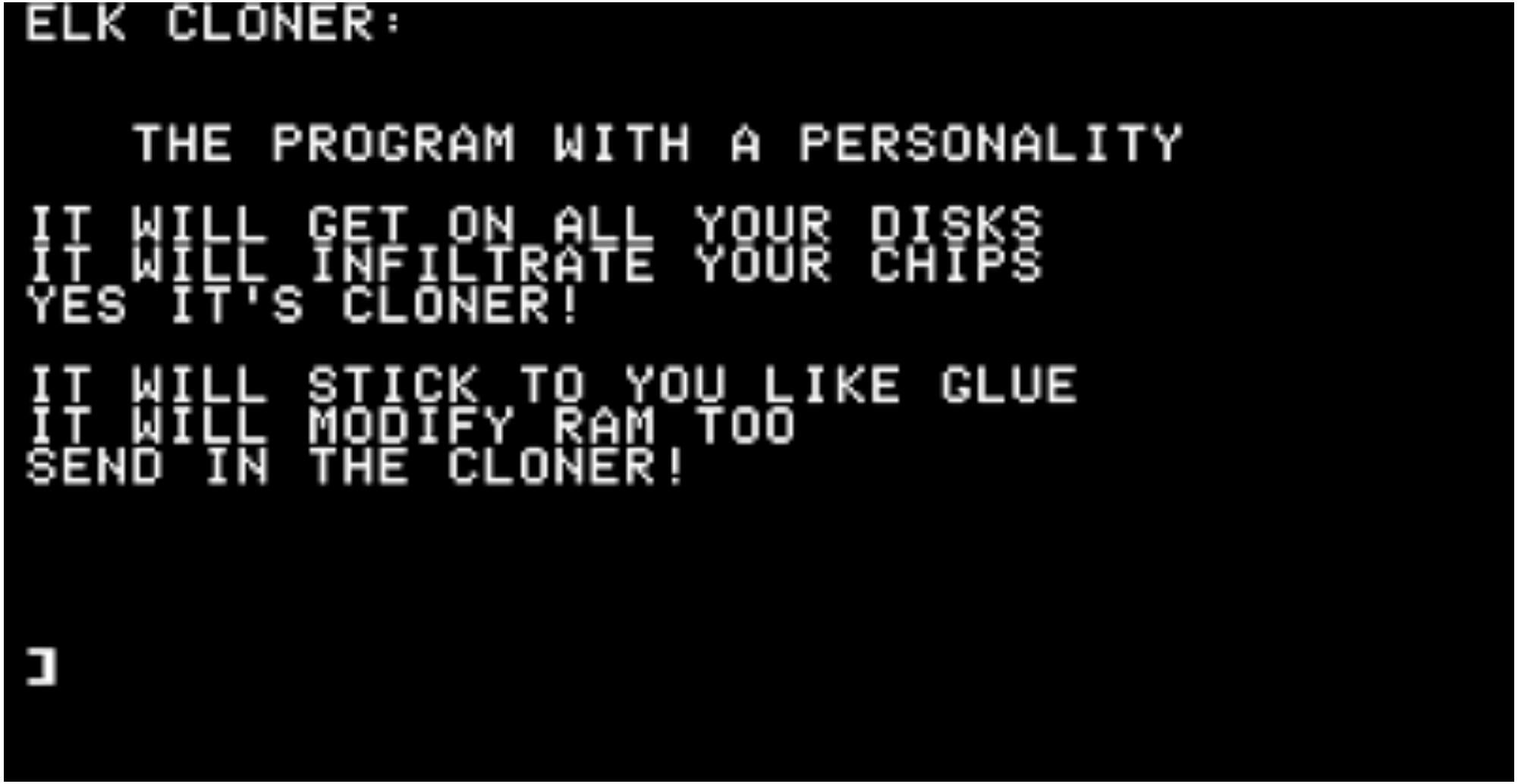
- Robert Moskovitch, Clint Feher, and Yuval Elovici. 2009. A chronological evaluation of unknown malcode detection. *LNCS: Intelligence and Security Informatics* 5477 (2009), 112–117.
- Xin Hu. 2011. *Large-scale malware analysis, detection, and signature generation.* Ph.D. Dissertation, Department of Computer Science and Engineering, University of Michigan.
- Damballa. 2008. 3% to 5% of Enterprise Assets Are Compromised by Bot-Driven Targeted Attack Malware. Retrieved from http://www.prnewswire.com/news-releases/3-to-5-of-enterprise-assets-are-compromisedby-bot-driven-targeted-attack-malware-61634867.html.
- Yanfang Ye, Tao Li, Shenghuo Zhu, Weiwei Zhuang, Egemen Tas, Umesh Gupta, and Melih Abdulhayoglu. 2011. Combining file content and file relations for cloud based malware detection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.
- Isabelle Guyon and Andr Elisseeff. 2003. An introduction to variable and feature selection. *Jouranl of Machine Learning Research* 3 (March 2003), 1157–1182.
- Pat Langley. 1994. Selection of relevant features in machine learning. In *Proceedings of AAAI Fall Symposium*. Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 8 (2005), 1226–1238.
- J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* San Francisco, CA: Morgan Kaufmann Publishers, Inc. (1993).

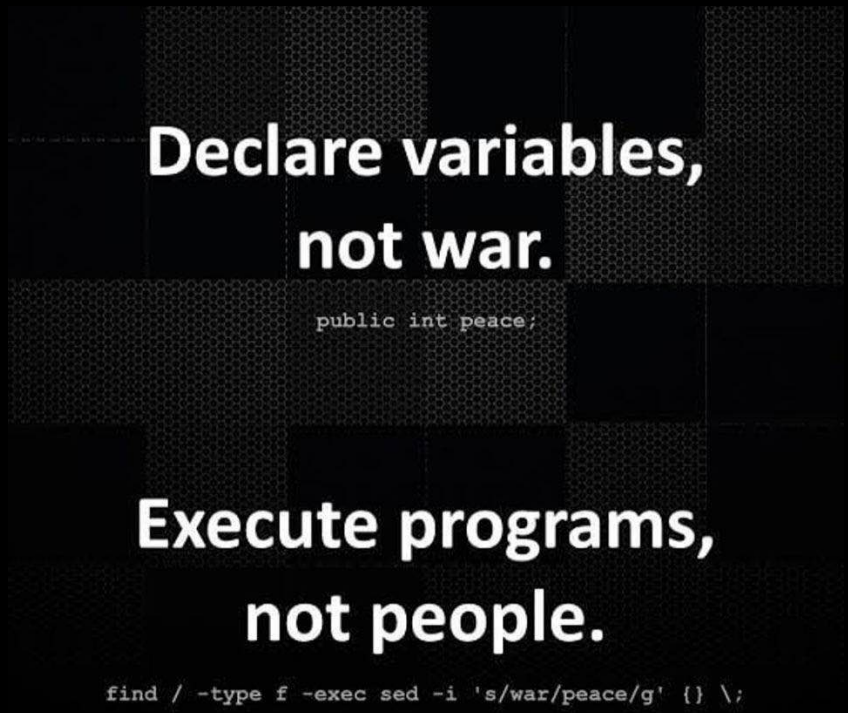
… Bibliography…

- George H. John and Pat Langley. 1995. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
- Pedro Domingos and Michael Pazzani. 1997. On the optimality of simple Bayesian classifier under zero-one loss. Machine Learning 29, 2–3 (1997), 103–130.
- Evelyn Fix and Joseph L. Hodges Jr. 1951. Discriminatory analysis-nonparametric discrimination: Consistency properties. *US Air Force, School of Avaiation Medicine, Tech. Rep* 4 (1951), 5–32.
- Christopher M. Bishop. 1995. Neural networks for pattern recognition. *Oxford, Clarendon Press.*
- Thorsten Joachims. 1998. Making large-scale support vector machine learning practical. *Advances in Kernel Methods: Support Vector Machines* (1998).
- Fadi Abdeljaber Thabtah. 2007. A review of associative classification mining. *Knowledge Engineering Review* 22, 1 (2007), 37–65.
- Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2, 1 (2009), 1–127.
- Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems* 19 (2007).
- Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems*.
- Ucci, D., Aniello, L., & Baldoni, R. (2018). Survey of machine learning techniques for malware analysis. *Computers & Security*
- Souri, Alireza, and Rahil Hosseini. "A state-of-the-art survey of malware detection approaches using data mining techniques." *Human-centric Computing and Information Sciences* 8, no. 1 (2018): 3.

… Bibliography

- Hu, W. and Tan, Y., 2017. Generating adversarial malware examples for black-box attacks based on GAN. *arXiv preprint arXiv:1702.05983*.
- Stefan Axelsson. The Base-Rate Fallacy and the Difficulty of Intrusion Detection. ACM TISSEC, 2000.
- Enrico Mariconti, Lucky Onwuzurike, Panagiotis Andriotis, Emiliano De Cristofaro, Gordon Ross, and Gianluca Stringhini. MaMaDroid: Detecting Android Malware by Building Markov Chains of Behavioral Models. In NDSS, 2017.
- Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, and Konrad Rieck. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. In NDSS, 2014.
- Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J. and Cavallaro, L., 2019. {TESSERACT}: Eliminating Experimental Bias in Malware Classification across Space and Time. In *28th {USENIX} Security Symposium ({USENIX} Security 19)* (pp. 729-746).


ELK CLONER:
THE PROGRAM WITH A PERSONALITY
IT WILL GET ON ALL YOUR DISKS
IT WILL INFILTRATE YOUR CHIPS
YES IT'S CLONER!
IT WILL STICK TO YOU LIKE GLUE
IT WILL MODIFY RAM TOO
SEND IN THE CLONER!
]


Declare variables,
not war.
public int peace;
Execute programs,
not people.
find / -type f -exec sed -i 's/war/peace/g' {} \;