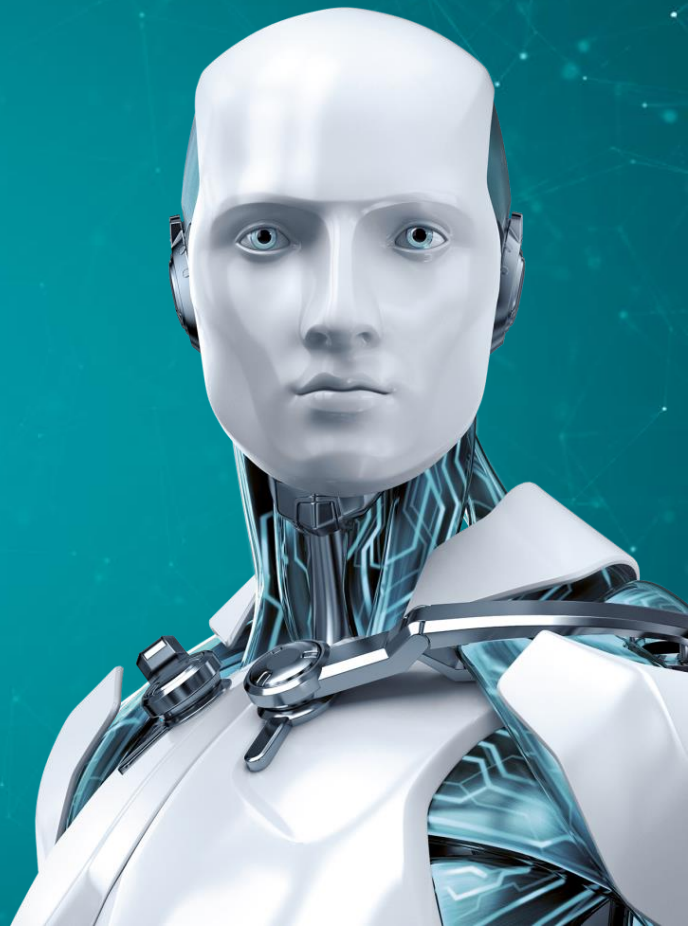
ENJOY SAFER TECHNOLOGY™

LATAM financial cybercrime

Competitors in crime sharing TTPs



Jakub Souček | Malware Analyst

Martin Jirkal | PRG Analyst Team Lead

LATAM banking trojans

- Dominate LATAM cybercrime
- ESET's research since 2016
- Identified 11(+2) distinct families
- Talk at Botconf 2019

ENJOY SAFER TECHNOLOGY™

LATAM banking trojans

- Dominate LATAM cybercrime
- ESET's research since 2016
- Identified 11(+2) distinct families
- Talk at Botconf 2019
- Started seeing similarities

- Dominat
- ESET’s re
- Identifie
- Talk at B
- Started s

"can I copy your homework?"

"yeah just change it up a bit so it doesn't look obvious you copied"

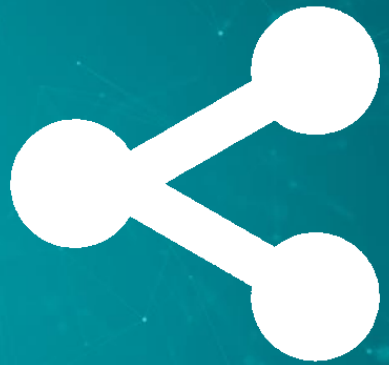
"ok"

LATAM banking trojans

- Dominate LATAM cybercrime
- ESET's research since 2016
- Identified 11(+2) distinct families
- Talk at Botconf 2019
- Started seeing similarities
- We will share these similarities with you

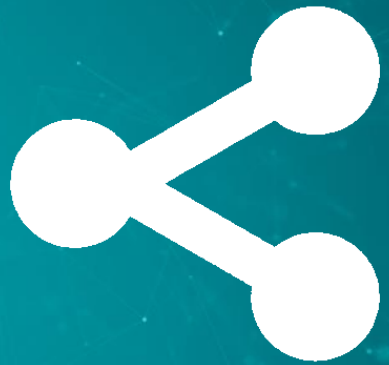
What is shared between LATAM banking trojans?

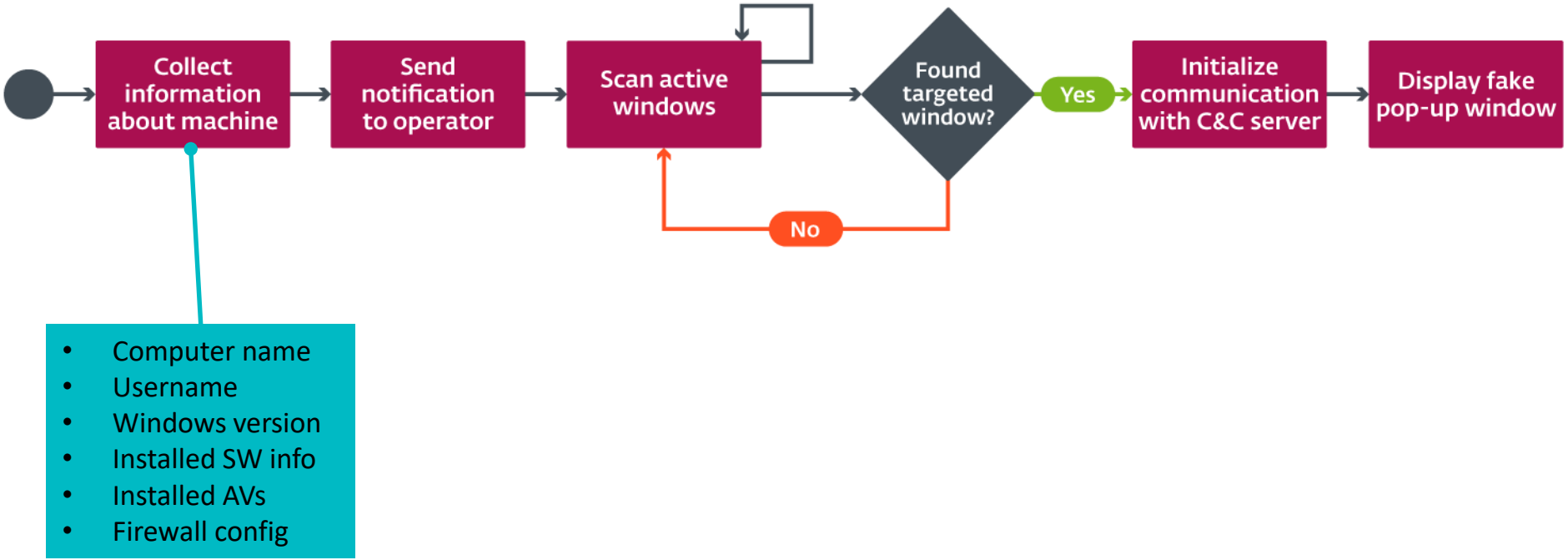
- Binary characteristics
- Distribution chains
- Execution methods
- Geographical distribution

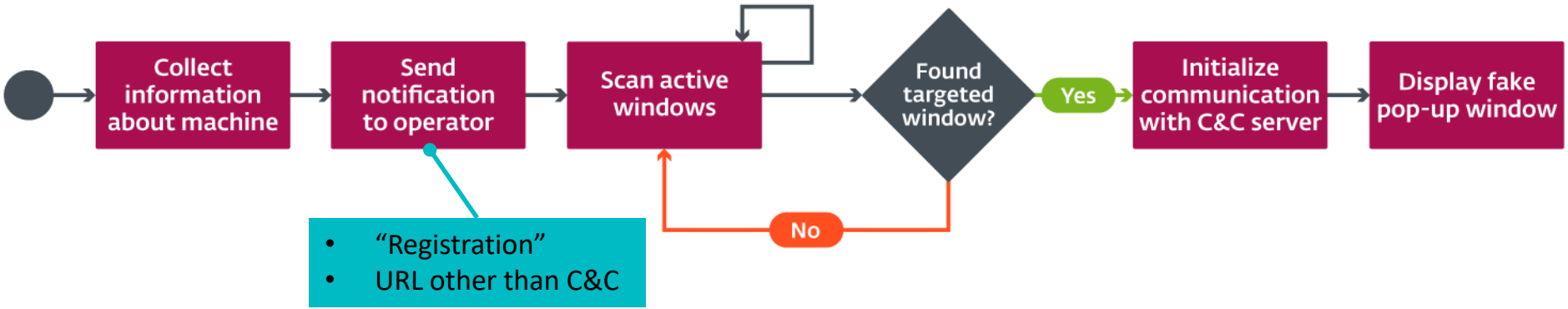


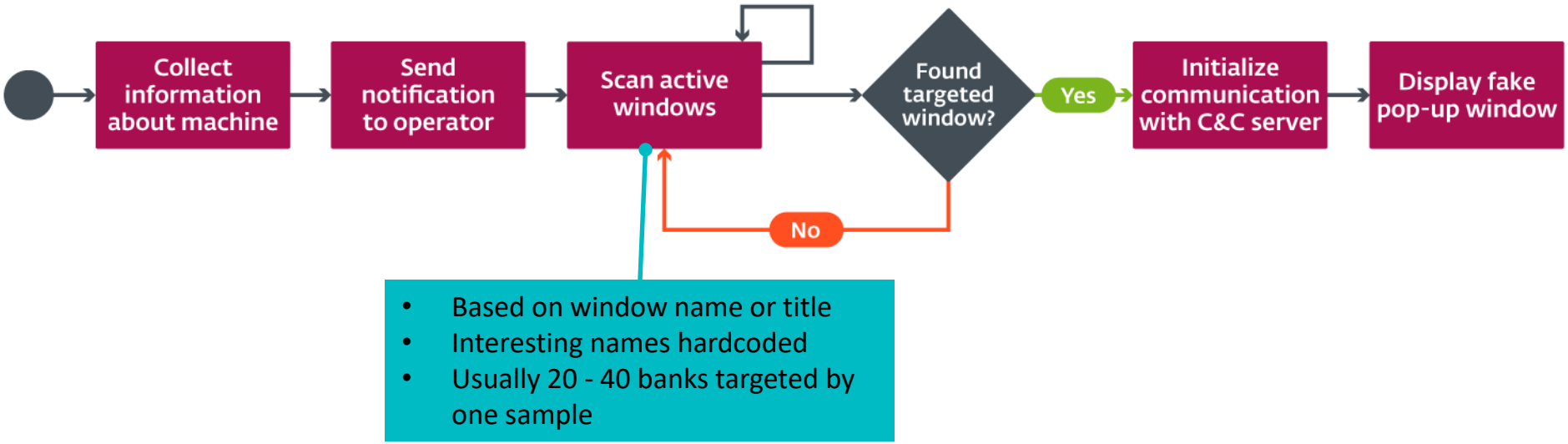
What is shared between LATAM banking trojans?

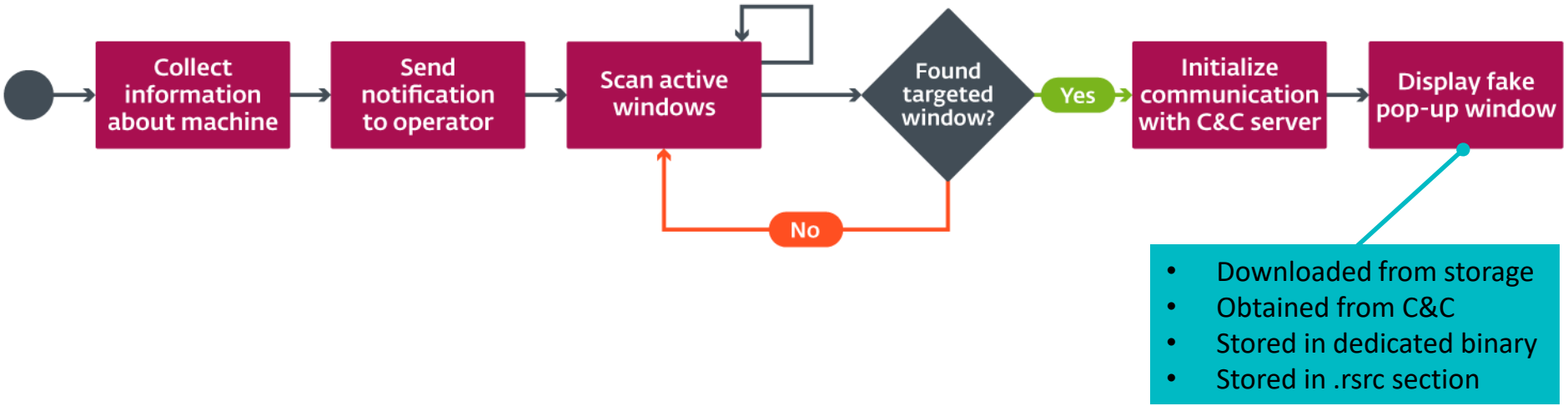
- Binary characteristics
- Distribution chains
- Execution methods
- Geographical distribution

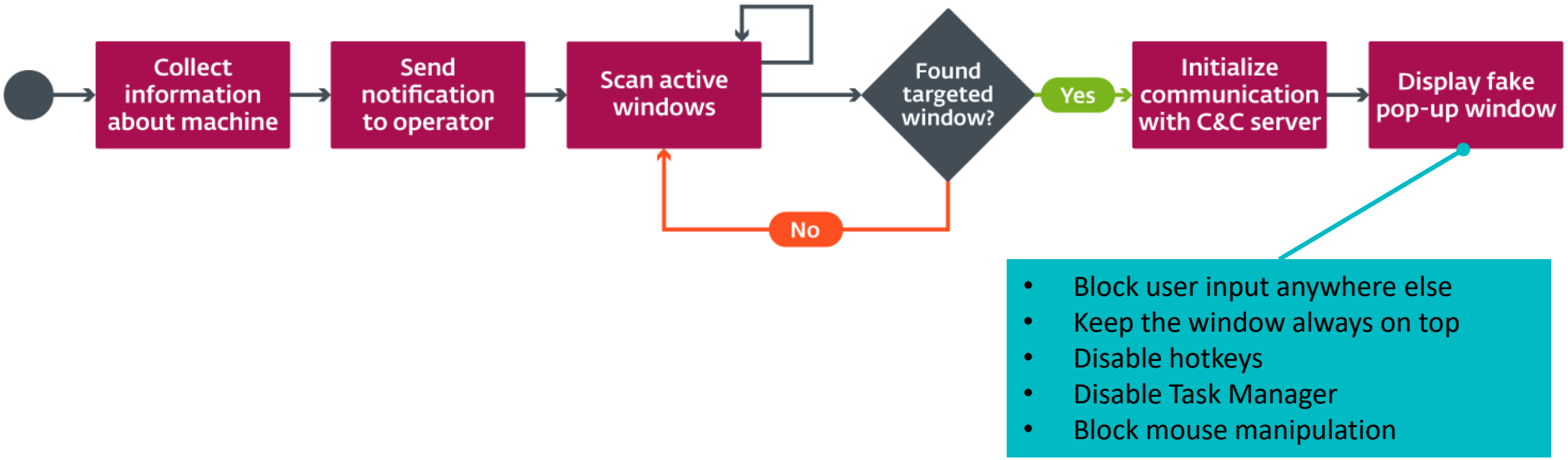



Collect information about machine
Send notification to operator
Scan active windows
Found targeted window?
Yes
No
Initialize communication with C&C server
Display fake pop-up window
Computer name
Username
Windows version
Installed SW info
Installed AVs
Firewall config


Collect information about machine
Send notification to operator
Scan active windows
Found targeted window?
Yes
No
Initialize communication with C&C server
Display fake pop-up window
• "Registration"
• URL other than C&C


Collect information about machine
Send notification to operator
Scan active windows
Found targeted window?
Yes
No
Initialize communication with C&C server
Display fake pop-up window
• Based on window name or title
• Interesting names hardcoded
• Usually 20 - 40 banks targeted by one sample


Collect information about machine
Send notification to operator
Scan active windows
Found targeted window?
Yes
No
Initialize communication with C&C server
Display fake pop-up window
• Downloaded from storage
• Obtained from C&C
• Stored in dedicated binary
• Stored in .rsrc section

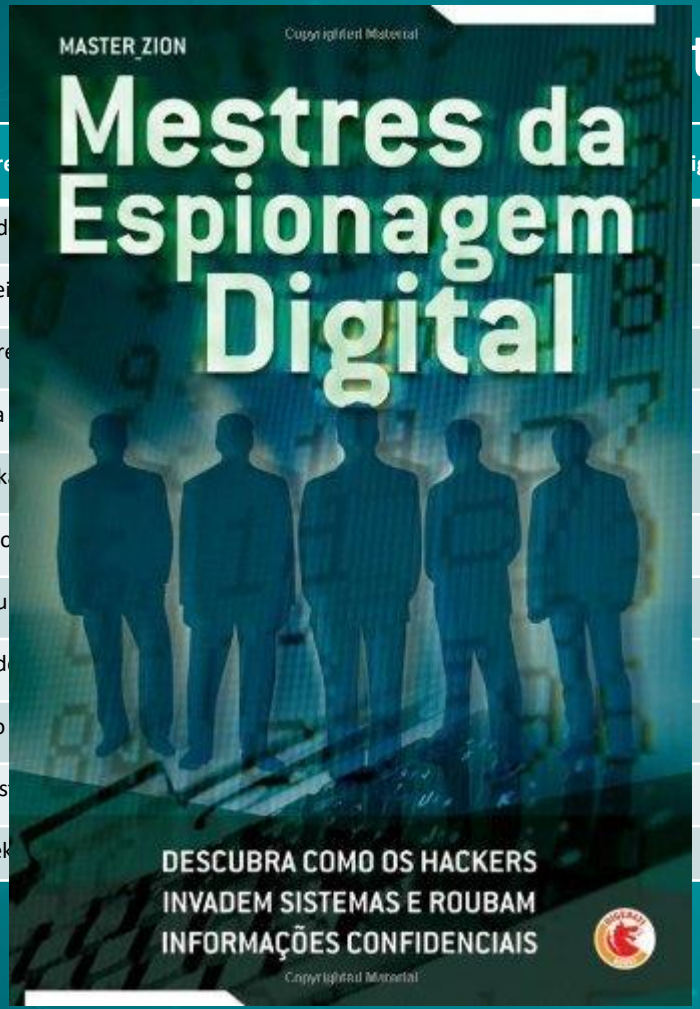

Collect information about machine
Send notification to operator
Scan active windows
Found targeted window?
Yes
No
Initialize communication with C&C server
Display fake pop-up window
• Block user input anywhere else
• Keep the window always on top
• Disable hotkeys
• Disable Task Manager
• Block mouse manipulation

String Encryption

Malware family	TripleKey	BookDecrypt	XOR_FF	KeySub	BigAlpha	Division
Amavaldo	✓					
Casbaneiro	✓	✓				
Grandoreiro		✓	✓			
Guildma		✓			✓	
Krachulka						
Lokorrito				✓		
Mispadu					✓	
Numando		✓				✓
Mekotio		✓				
Vadokrist	✓		✓			
Zumanek		✓				

tion

Malware	igAlpha	Division
Amavald		
Casbanei		
Grandore		
Guildma	✓	
Krachulk		
Lokorrito		
Mispadu	✓	
Numand		✓
Mekotio		
Vadokris		
Zumanek		



String table

```
mov     dl, 1
mov     eax, VMT_1A8C2B8_TStringList ; TStringList_Self
call    TStringList_Create
mov     [ebp+StringTable], eax
xor     eax, eax
push    ebp
push    offset loc_1CB1D43
push    dword ptr fs:[eax]
mov     fs:[eax], esp
mov     edx, offset a55987aa7509473 ; "55987AA7509473 9A3E5DF746C3"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a8f819c4af958bb ; "8F819C4AF958BB56F60B4F858294B69E"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a538b78aa5da7a1 ; "538B78AA5DA7A1A82BAAE2728AC389AA5CF95EB"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a9ec152f118b463 ; "9EC152F118B463CF7BED529F66FA"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
xor     edx, edx        ; string_S
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a34808aa95be753 ; "34808AA95BE753"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset aE7050c3ce368a0 ; "E7050C3CE368A06BB2CF0C45CE030A"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset aDc0f09719232e8 ; "DC0F09719232E86FE86A"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
...
mov     eax, [ebp+StringTable]
mov     esi, [eax]
call    [esi+TStringList.TStringList_Get_0xc]
mov     eax, [ebp+EncryptedString]
mov     edx, ebx
call    DecryptString
xor     eax, eax
pop     edx
pop     ecx
pop     ecx             ; this
```

```
mov     dl, 1
mov     eax, ds:VMT_496580_TStringList ; TStringList_Self
call    TStringList_Create
mov     [ebp+StringTable], eax
xor     eax, eax
push    ebp
push    offset loc_6307E3
push    dword ptr fs:[eax]
mov     fs:[eax], esp
mov     edx, offset a11865c4af9be5b ; "11865C4AF9BE5B439AAD050FD84110F5866B510"...
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a1925b881ff4988 ; "1925B881FF498896F1223DE319792098BAE0"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a24e73c157e1e92 ; "24E73C157E1E92C70AFDB0"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3dd1cb3ad02cdf ; "3DD1CB3AD02CDF46E918CC3C08F7"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3508 ; "3508"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a396c503ebec799 ; "396C503EBEC7999031F30FC1B288C13E73"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3947 ; "3947"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3dd743a2433601 ; "3DD743A2433601ED9247E5FD62DF5A"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
...
mov     eax, [ebp+StringTable]
mov     esi, [eax]
call    [esi+TStringList.TStringList_Get_0xc]
mov     eax, [ebp+EncryptedString]
mov     edx, ebx
call    DecryptString
xor     eax, eax
pop     edx
pop     ecx
pop     ecx             ; this
```

String table

```
mov     dl, 1
mov     eax, UMT_1A8C2B8_TStringList ; TStringList_Self
call    TStringList_Create
mov     [ebp+StringTable], eax
xor     eax, eax
push    ebp
push    offset loc_1CB1D43
push    dword ptr fs:[eax]
mov     fs:[eax], esp
mov     edx, offset a55987aa7509473 ; "55987AA7509473 9A3E5DF746C3"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a8f819c4af958bb ; "8F819C4AF958BB56F60B4F858294B69E"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a538b78aa5da7a1 ; "538B78AA5DA7A1A82BAAE2728AC389AA5CF95EB"...
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a9ec152f118b463 ; "9EC152F118B463CF7BED529F66FA"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
xor     edx, edx        ; string_S
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a34808aa95be753 ; "34808AA95BE753"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset aE7050c3ce368a0 ; "E7050C3CE368A06BB2CF0C45CE030A"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset aDc0f09719232e8 ; "DC0F09719232E86FE86A"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
...
mov     eax, [ebp+StringTable]
mov     esi, [eax]
call    [esi+TStringList.TStringList_Get_0xc]
mov     eax, [ebp+EncryptedString]
mov     edx, ebx
call    DecryptString
xor     eax, eax
pop     edx
pop     ecx
pop     ecx             ; this
```

Casbaneiro

```
mov     dl, 1
mov     eax, ds:UMT_496580_TStringList ; TStringList_Self
call    TStringList_Create
mov     [ebp+StringTable], eax
xor     eax, eax
push    ebp
push    offset loc_6307E3
push    dword ptr fs:[eax]
mov     fs:[eax], esp
mov     edx, offset a11865c4af9be5b ; "11865C4AF9BE5B439AAD050FD84110F5866B510"...
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a1925b881ff4988 ; "1925B881FF498896F1223DE319792098BAE0"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a24e73c157e1e92 ; "24E73C157E1E92C70AFDB0"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3dd1cb3ad02cdf ; "3DD1CB3AD02CDF46E918CC3C08F7"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3508 ; "3508"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a396c503ebec799 ; "396C503EBEC7999031F30FC1B288C13E73"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3947 ; "3947"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
mov     edx, offset a3dd743a2433601 ; "3DD743A2433601ED9247E5FD62DF5A"
mov     eax, [ebp+StringTable] ; TStringList_Self
mov     ecx, [eax]
call    [ecx+TStringList.TStringList_Add_0x3c]
...
mov     eax, [ebp+StringTable]
mov     esi, [eax]
call    [esi+TStringList.TStringList_Get_0xc]
mov     eax, [ebp+EncryptedString]
mov     edx, ebx
call    DecryptString
xor     eax, eax
pop     edx
pop     ecx
pop     ecx             ; this
```

Vadokrist

Implementation details

- ALL written in Delphi
- Large binaries
- Delphi_Remote_Access_PC
  - Amavaldo, Casbaneiro, Mekotio, Mispadu, Vadokrist
- Magnification.dll
  - Vast majority of the families
  - For taking screenshots
- VMProtect, Themida
- Disabling Google Chrome hardware acceleration

Simulate click of RadioButton in Google Chrome with SendMessage

Asked 4 years, 9 months ago Active 4 years, 9 months ago Viewed 398 times

1

I have an application that performs clicks in the Google chrome browser through `sendmessage`. It works perfectly when I click on all components of the page without problem with a single exception: I can not select a RadioButton.

This is the only part I can not click:

I have already tried the PostMessageA And SendNotifyMessage alternatives without success, and I also found code on the internet that supposedly solved the problem, but no success - I've included that code below.

Observation: it would not be feasible using conventional click system for my application.

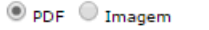
Simulate click of RadioButton in Google Chrome with SendMessage

Asked 4 years, 9 months ago Active 4 years, 9 months ago Viewed 398 times

1

I have an application that performs clicks in the Google ch
works perfectly when I click on all components of the pag
I can not select a RadioButton.

This is the only part I can not click:



I have already tried the PostMessageA And SendNotifyMe
also found code on the internet that supposedly solved th
that code below.

Observation: it would not be feasible using conventional c

Code that promises to solve the problem however did not work

```
procedure TForm1.ChromeBugFix;
var
  texto: string;
  specialfolder, I: integer;
  ARQ: TStringList;
  Ln, NewLn: String;
  caminhochrome: String;
begin
 specialFolder := CSIDL_LOCAL_APPDATA;
 caminhochrome := GetSpecialFolderPath(specialFolder);
 caminhochrome := caminhochrome + '\Google\Chrome\User Data\Local State';
  if fileexists(caminhochrome) then
  begin
   ARQ:=TStringList.Create;
   ARQ.LoadFromFile(caminhochrome);
   ARQ.Text:=StringReplace(ARQ.Text,'"enabled": true','"enabled": false',[rfReplaceAll]);
   ARQ.Text:=StringReplace(ARQ.Text,'"hardware_acceleration_mode_previous": true','"hardwar
    for I := 0 to ARQ.Count - 1 do
    begin
      ARQ[0] := '{ "hardware_acceleration_mode": { "enabled": false },';
    end;
     ARQ.SavetoFile(caminhochrome);
  end;
end;
```

Simulate click of RadioButton in Google Chrome with SendMessage

Asked 4 years, 9 months ago Active 4 years, 9 months ago Viewed 398 times

1

I have an application that performs clicks in the Google ch
works perfectly when I click on all components of the pag
I can not select a RadioButton.

This is the only part I can not click:

PDF Imagem

I have already tried the PostMessageA And SendNotifyMe
also found code on the internet that supposedly solved th
that code below.

Observation: it would not be feasible u

Code that promises to solve the problem however did not work

```
procedure TForm1.ChromeBugFix;
var
  texto: string;
  specialfolder, I: integer;
  ARQ: TStringList;
  Ln, NewLn: String;
  caminhochrome: String;
begin
 specialFolder := CSIDL_LOCAL_APPDATA;
 := GetSpecialFolderPath(specialFolder);
 := caminhochrome + '\Google\Chrome\User Data\Local State';
(caminhochrome) then

List.Create;
File(caminhochrome);
ringReplace(ARQ.Text,'"enabled": true','"enabled": false',[rfReplaceAll]);
ringReplace(ARQ.Text,'"hardware_acceleration_mode_previous": true','"hardwar
to ARQ.Count - 1 do

 '{ "hardware_acceleration_mode": { "enabled": false },';

File(caminhochrome);

  end;
end;
```

PHP Receiv Post

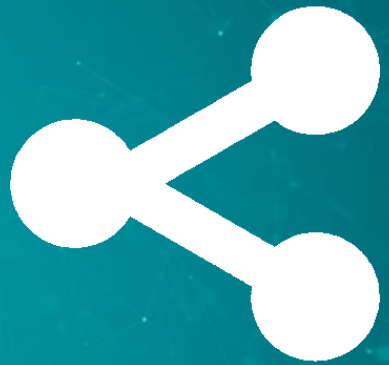
```
<?php

$cnpj = $_GET['cnpj'];
$cc = $_POST['chr_NumeroCartao'];
$mes = $_POST['int_MesCartao'];
$ano = $_POST['int_AnoCartao'];
$cvv =  $_POST['chr_CVC2'];

?>
```

Anti-fraud software

- Trusteer (IBM), Warsaw/GBPlugin (GAS Tecnologia)
- Reaction of LATAM banking trojans?
  - Discovery
    - report to attacker if installed
  - Protection
    - exit, hooking APIs
  - Disruption
    - file removal, firewall block, ACL modification, process kill

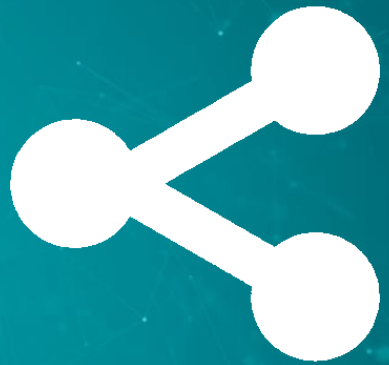
What is shared between LATAM banking trojans?

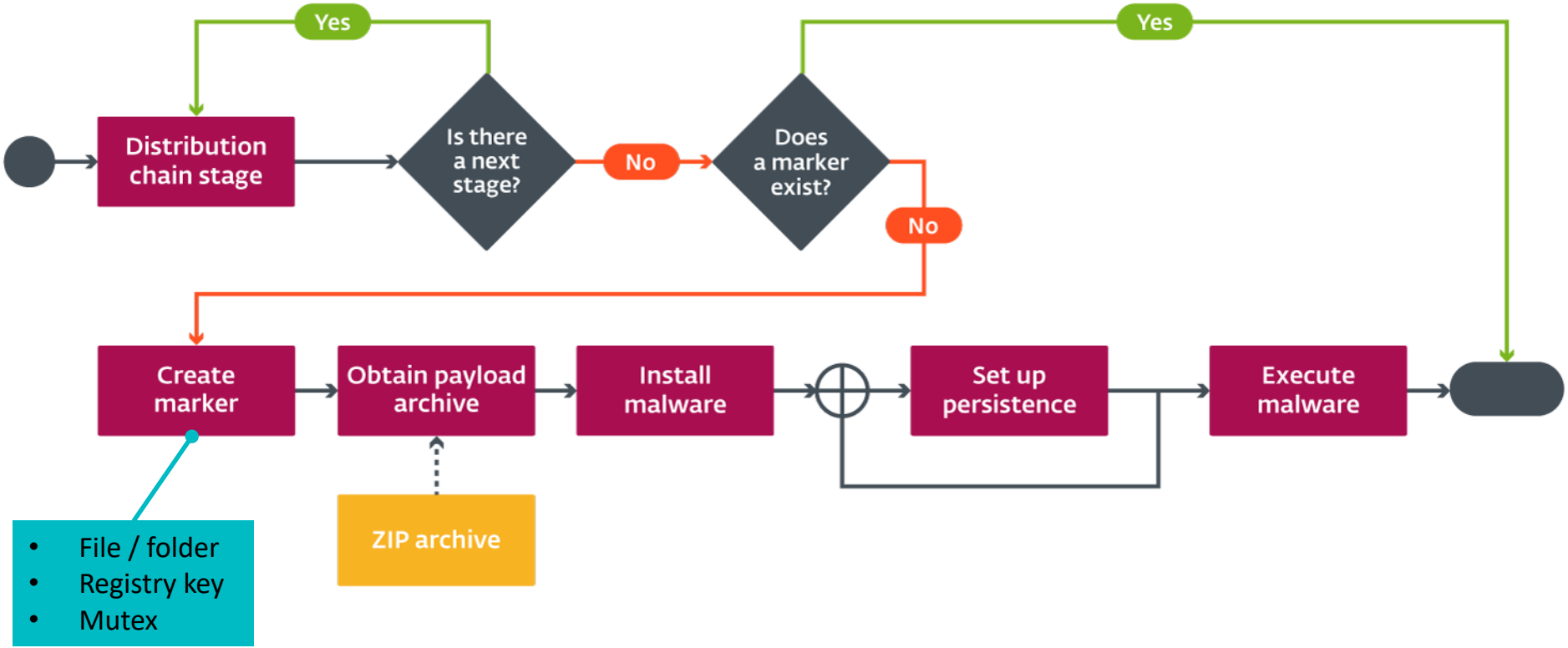
- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
- Execution methods
- Geographical distribution

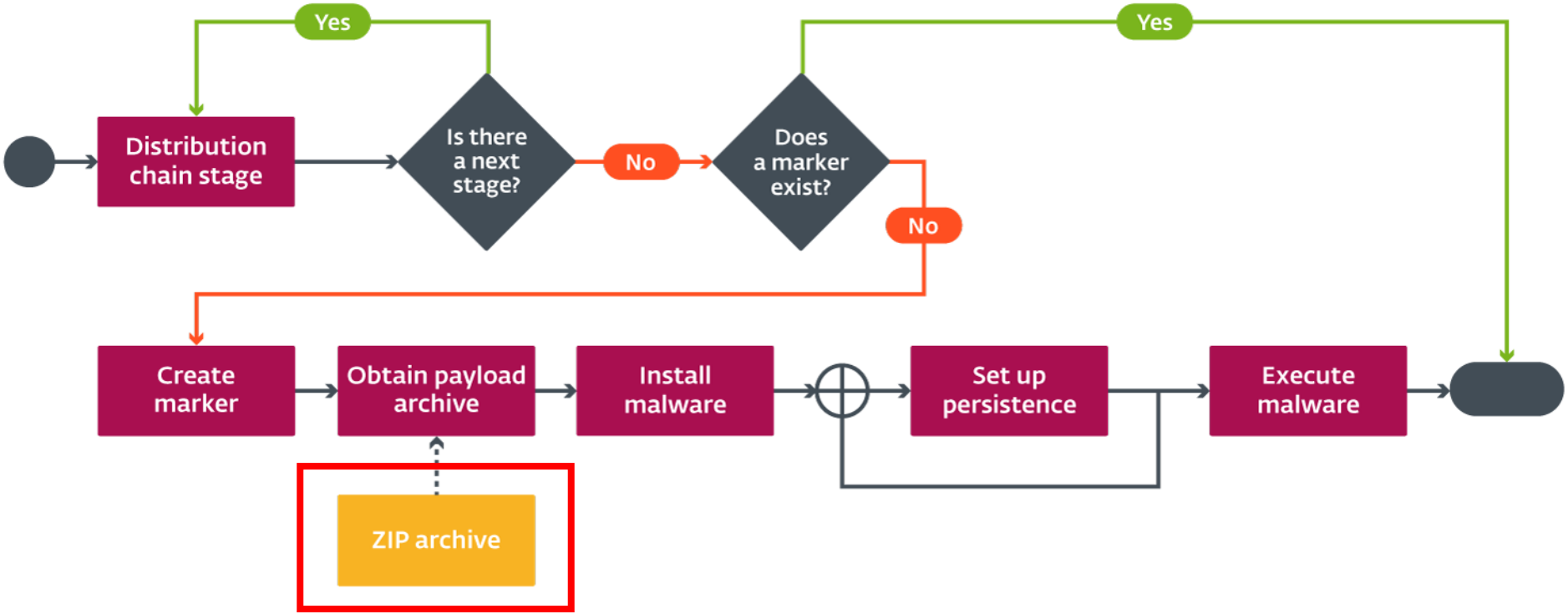


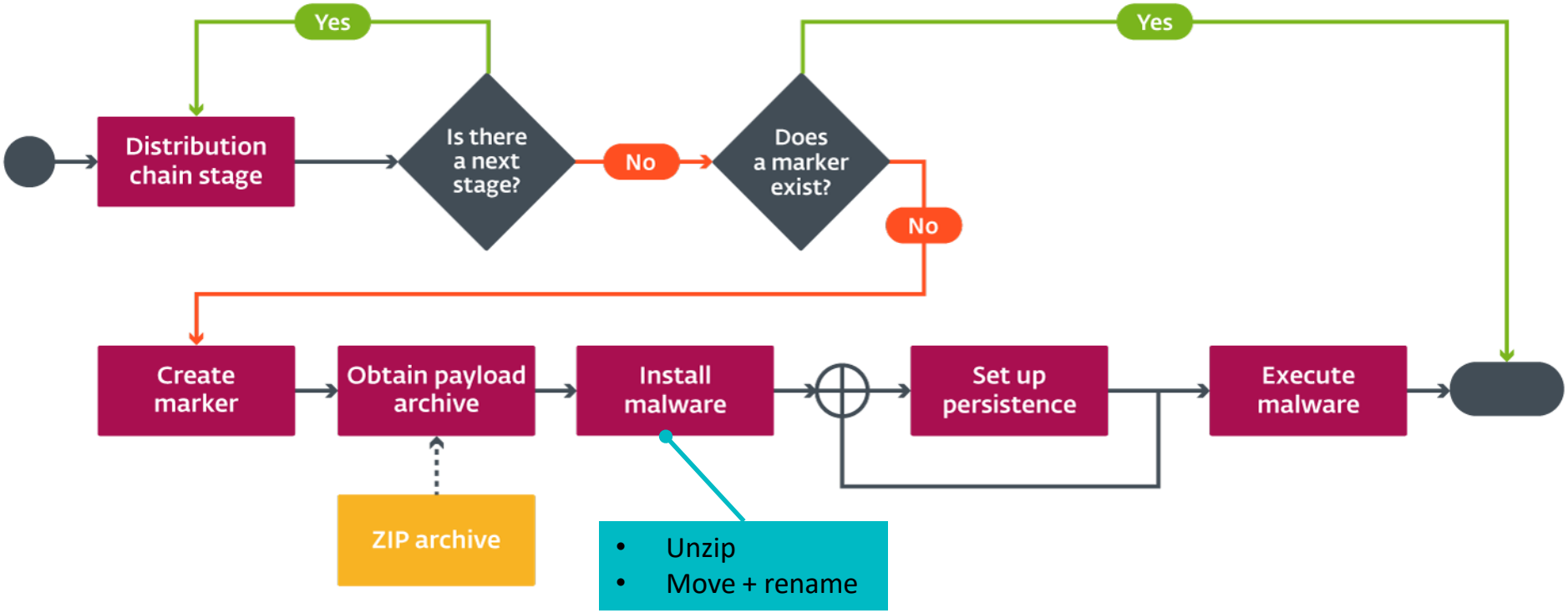
What is shared between LATAM banking trojans?

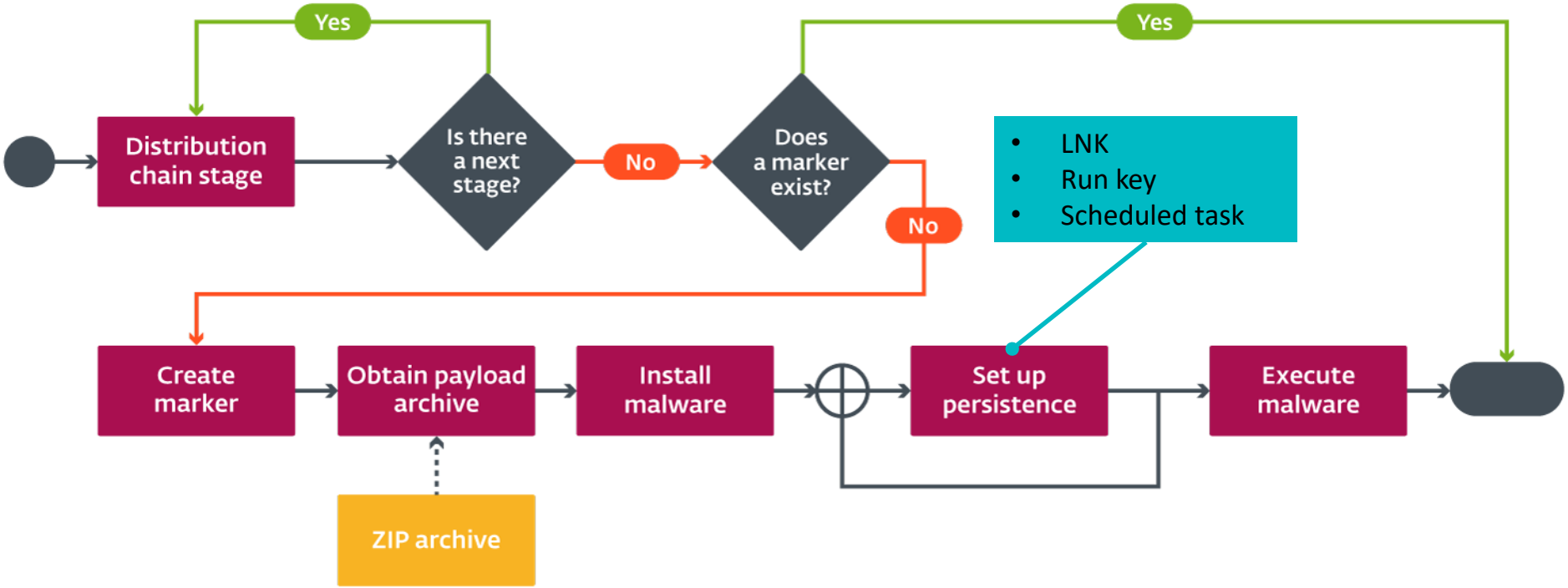
- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
- Execution methods
- Geographical distribution




Yes
Distribution chain stage
Is there a next stage?
No
Does a marker exist?
Yes
No
Create marker
Obtain payload archive
Install malware
Set up persistence
Execute malware
ZIP archive
• File / folder
• Registry key
• Mutex


Yes
Distribution chain stage
Is there a next stage?
No
Does a marker exist?
Yes
No
Create marker
Obtain payload archive
Install malware
Set up persistence
Execute malware
ZIP archive


Yes
Distribution chain stage
Is there a next stage?
No
Does a marker exist?
Yes
No
Create marker
Obtain payload archive
Install malware
Set up persistence
Execute malware
ZIP archive
• Unzip
• Move + rename


Yes
Distribution chain stage
Is there a next stage?
No
Does a marker exist?
Yes
No
• LNK
• Run key
• Scheduled task
Create marker
Obtain payload archive
Install malware
Set up persistence
Execute malware
ZIP archive

Distribution chains

- Delphi, JS, PS1, AU3, BAT, VBS
- Used exclusively for LATAM banking trojans
- Tightly connected to how the trojan is executed
- Seem to be maintained by author(s) of the banking trojans
- Every family has its own set…

Distribution chains

- … with some exceptions

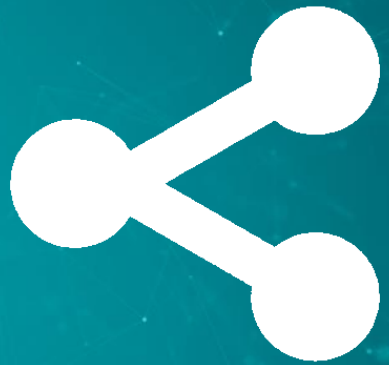
Chain ID	Language(s)	Stages	Casbaneiro	Grandoreiro	Mekotio	Vadokrist
1	Delphi	1	✓		✓	
2	Delphi	1		✓	✓	
3	Delphi	1		✓		✓
4	PowerShell	1	✓		✓	✓
5	JavaScript	1			✓	✓
6	BAT, VBScript, PowerShell	4			✓	✓

The first link in the chain

- LNK (in 2017, rarely used nowadays)
- HTML (currently used mainly by Guildma)
- MSI
  - Trending among LATAM banking trojans since 2019
  - Use of Advanced Installer to create an MSI that will
    1. Execute an embedded Delphi file or
    2. Download from an embedded URL and execute the response or
    3. Execute an embedded script

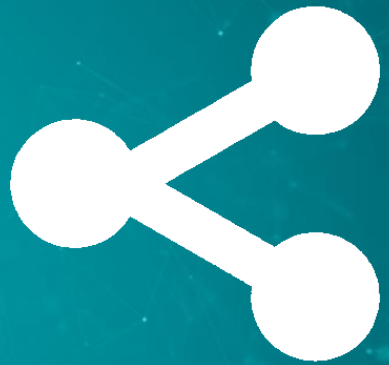
What is shared between LATAM banking trojans?

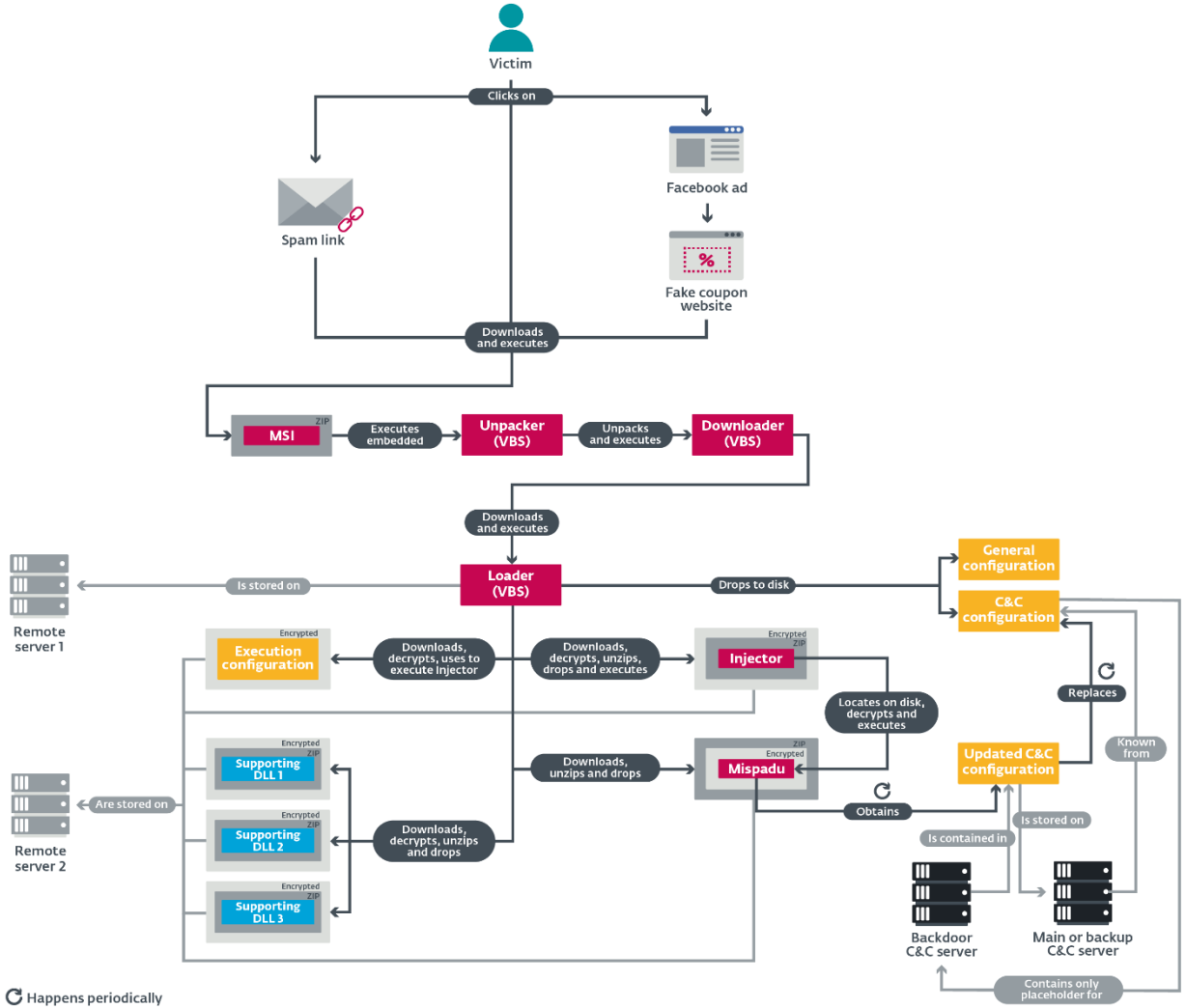
- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
  - Logic, specific chains, ZIP, MSI
- Execution methods
- Geographical distribution



What is shared between LATAM banking trojans?

- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
  - Logic, specific chains, ZIP, MSI
- Execution methods
- Geographical distribution



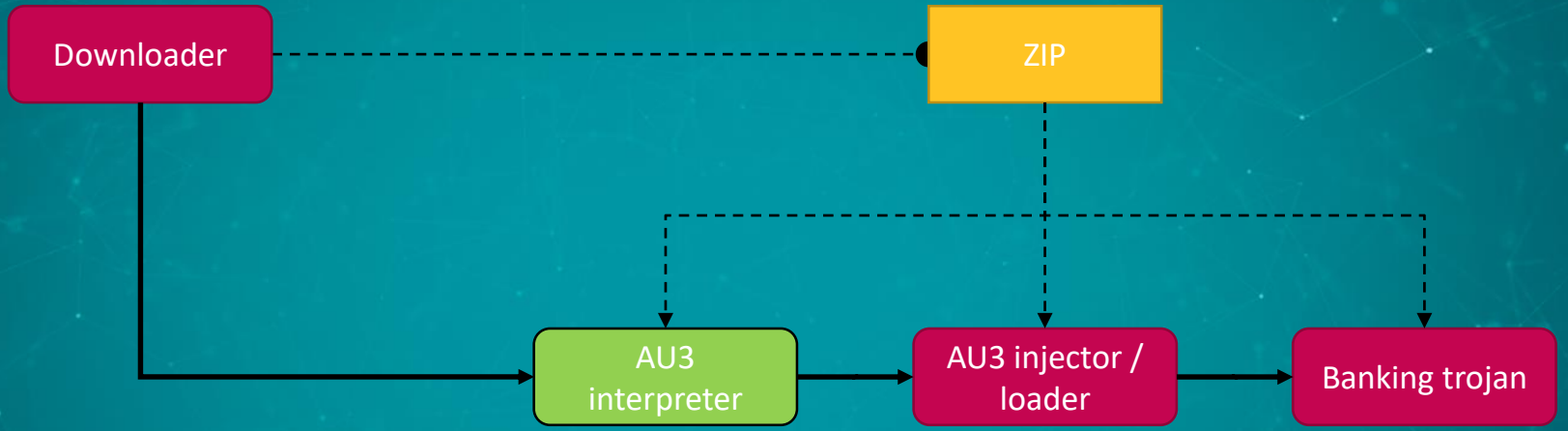

Victim
Clicks on
Spam link
Facebook ad
Fake coupon website
Downloads and executes
MSI
ZIP
Executes embedded
Unpacker (VBS)
Unpacks and executes
Downloader (VBS)
Downloads and executes
Loader (VBS)
Is stored on
Remote server 1
Drops to disk
General configuration
C&C configuration
Encrypted
Execution configuration
Downloads, decrypts, uses to execute injector
Downloads, decrypts, unzips, drops and executes
Encrypted
ZIP
Injector
Replaces
Locates on disk, decrypts and executes
Known from
Encrypted
ZIP
Supporting DLL 1
Encrypted
ZIP
Supporting DLL 2
Encrypted
ZIP
Supporting DLL 3
Are stored on
Remote server 2
Downloads, decrypts, unzips and drops
Downloads, unzips and drops
ZIP
Encrypted
Mispadu
Obtains
Updated C&C configuration
Is stored on
Is contained in
Backdoor C&C server
Main or backup C&C server
Contains only placeholder for
Happens periodically

How are LATAM banking trojans executed?

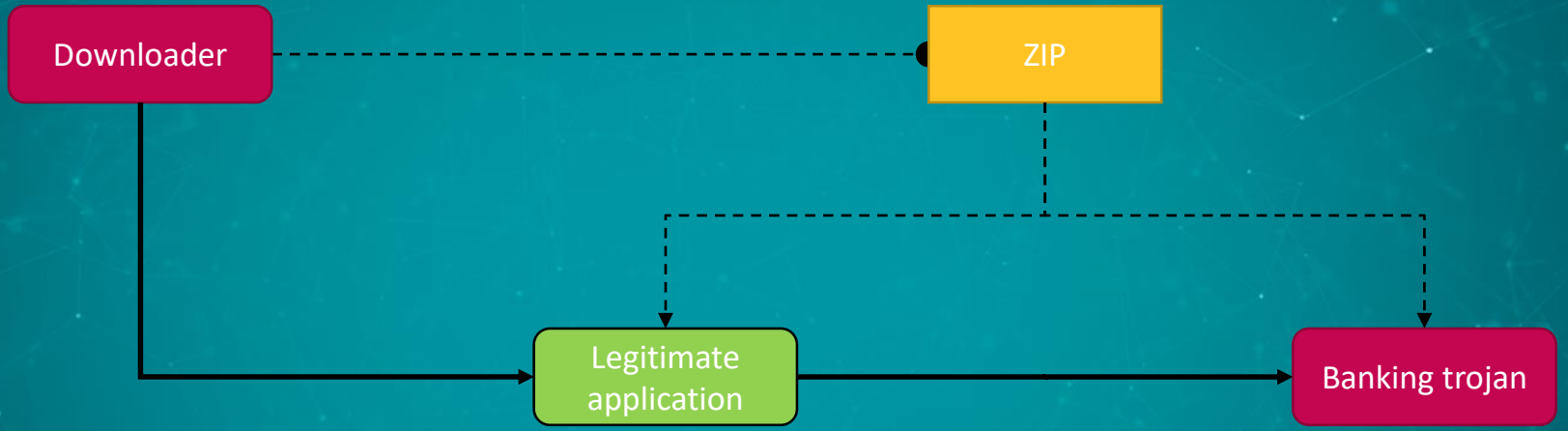
- Execution of distribution chain stages
  - LoLBins
    - BITSAdmin, Certutil, Msiexec, PowerShell, WMIC, WScript
- Execution of banking trojan
  - LoLBins
    - ExtExport, RegSvr32, RunDll32 (Guildma only)
  - Other families???

Direct

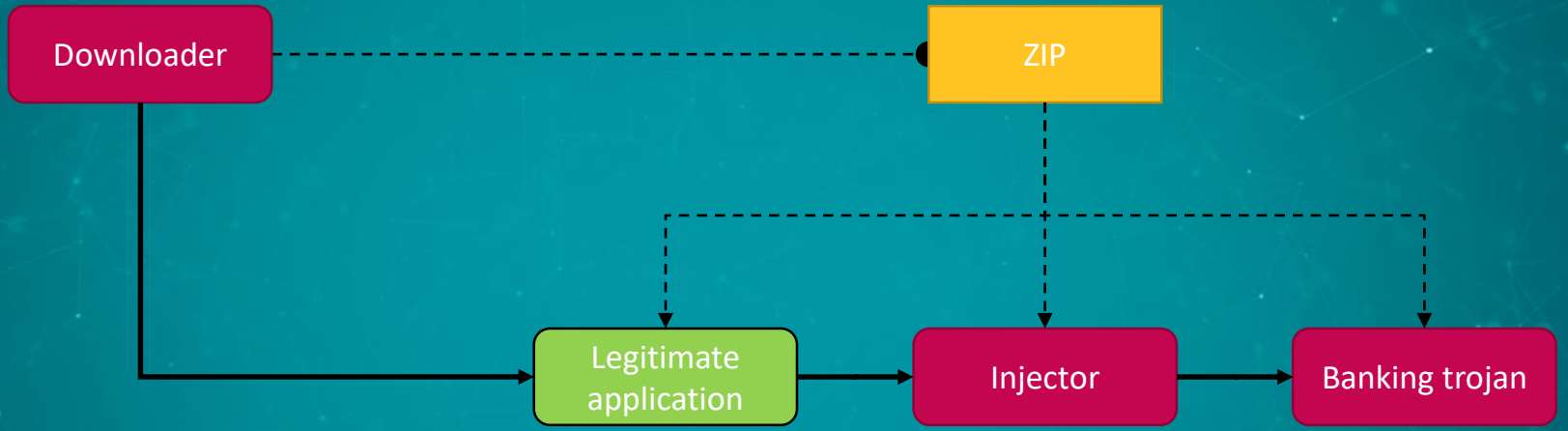
Abusing AutoIt



DLL side-loading



DLL side-loading with an injector



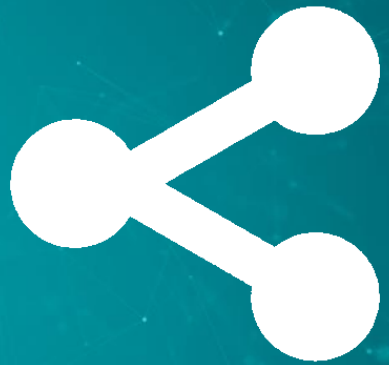
Execution methods

Method	Amavaldo	Casbaneiro	Lokorrito	Mekotio	Numando	Vadokrist	Zumanek
Direct		✓		✓			✓
Abusing AutoIt		✓		✓		✓	
DLL side-loading		✓	✓	✓	✓	✓	
DLL side-loading + injector	✓	✓		✓		✓	

DLL side-loading – what is being abused?

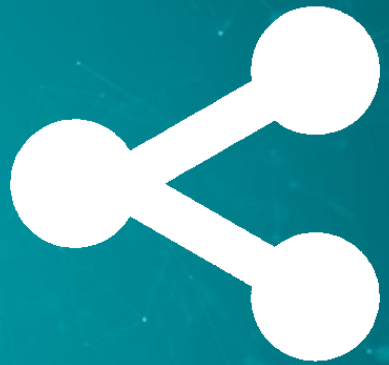
What is shared between LATAM banking trojans?

- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
  - Logic, specific chains, ZIP, MSI
- Execution methods
  - LoLBins, AU3 abusing, DLL side-loading, abused products
- Geographical distribution



What is shared between LATAM banking trojans?

- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
  - Logic, specific chains, ZIP, MSI
- Execution methods
  - LoLBins, AU3 abusing, DLL side-loading, abused products
- Geographical distribution

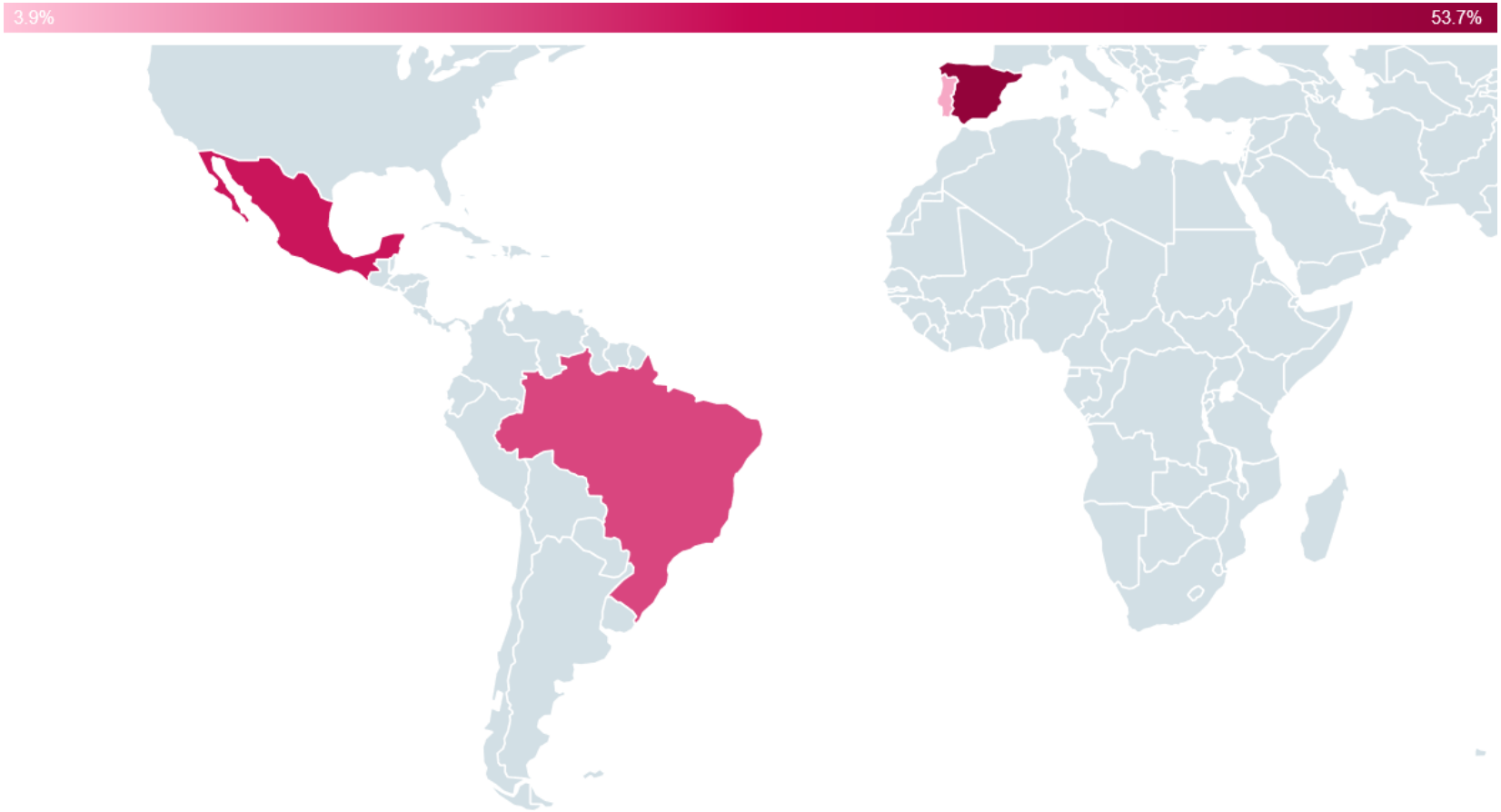


Geographical distribution

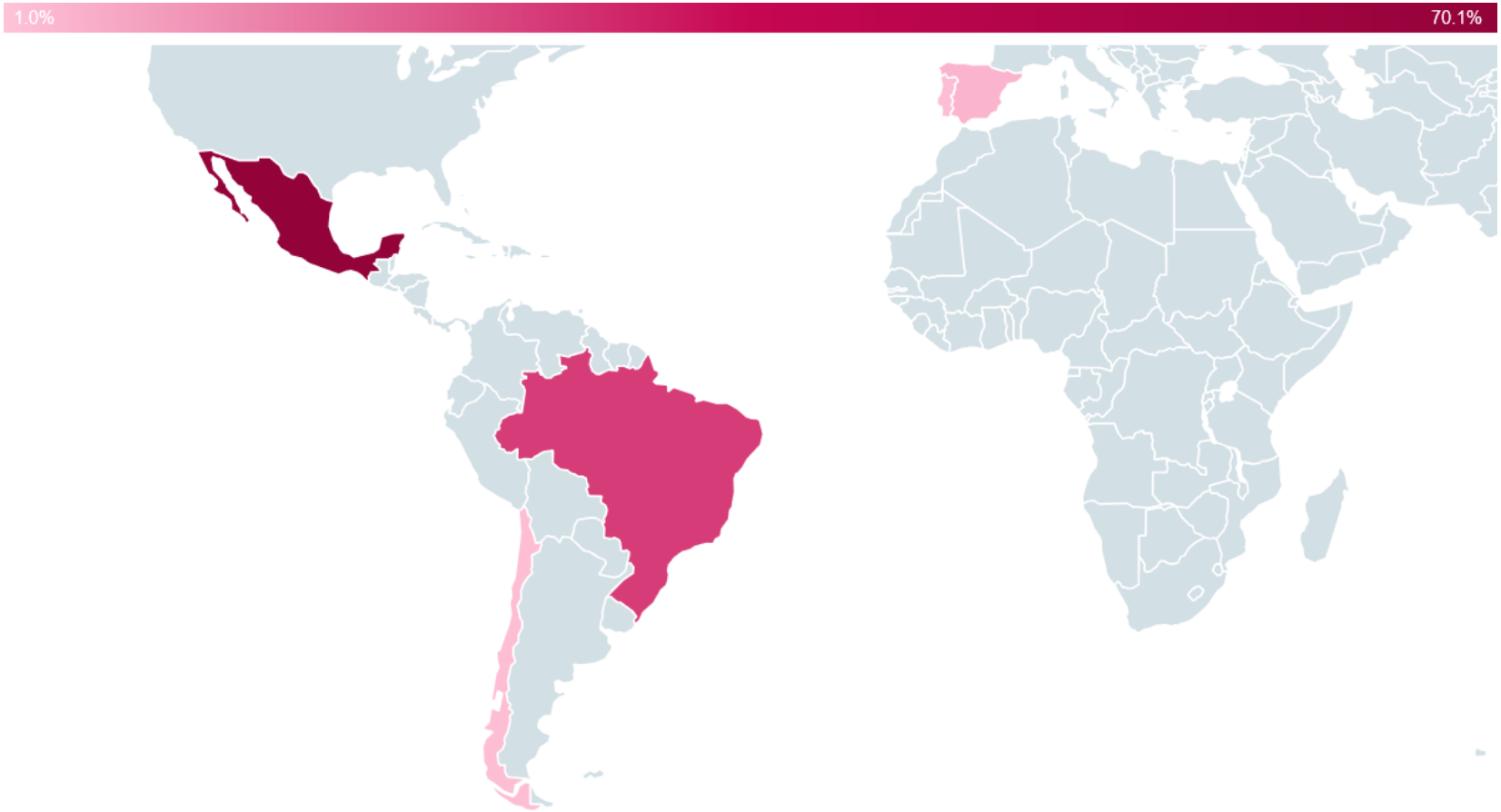
- Initially known to target Brazil
- Slowly affecting other countries (Perú, Chile)
- 2018/2019: Expansion to Mexico
- 2019/2020: Expansion to Spain and Portugal

Grandoreiro (before Oct 2019)

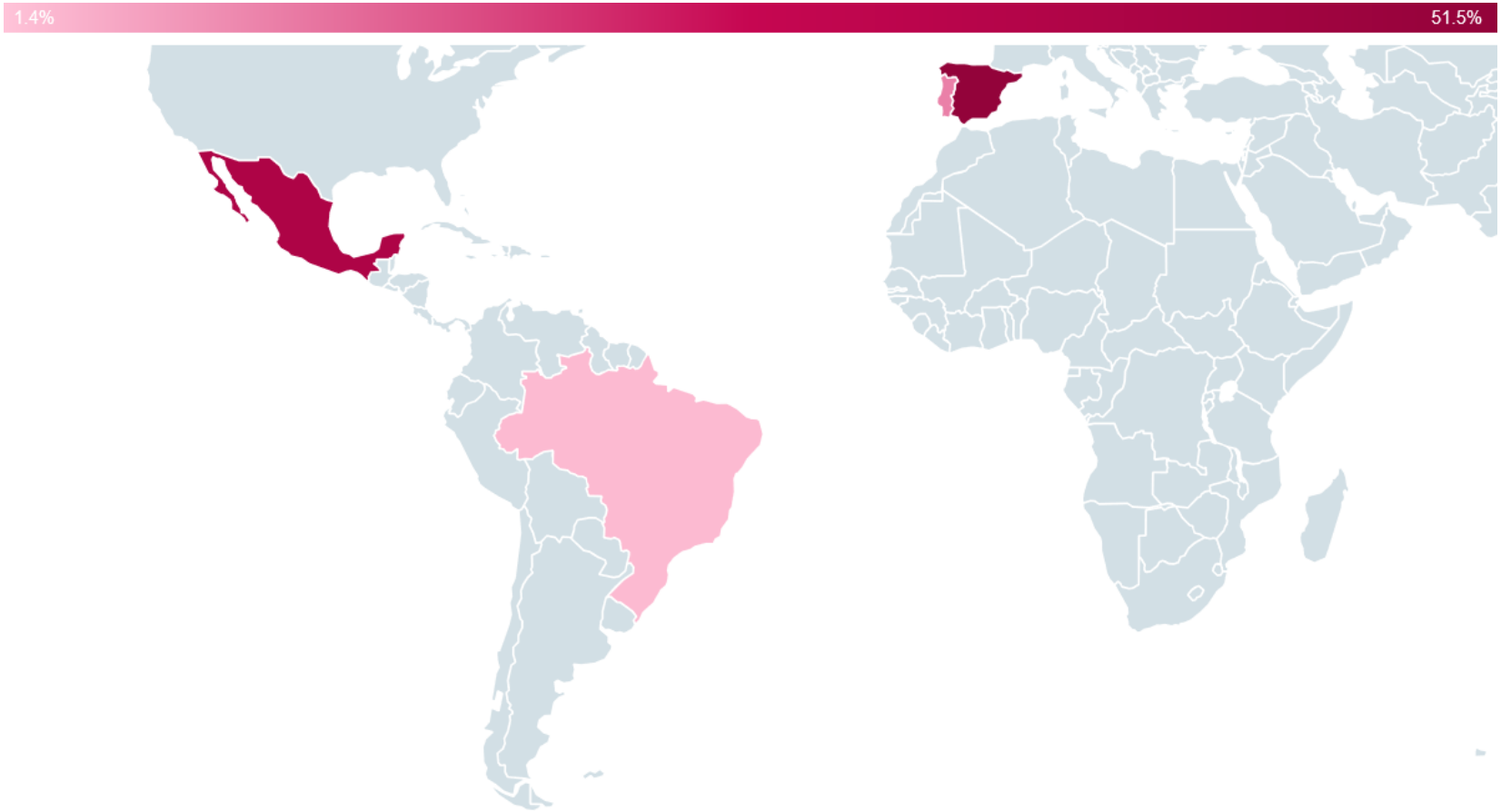
Grandoreiro (since Oct 2019)



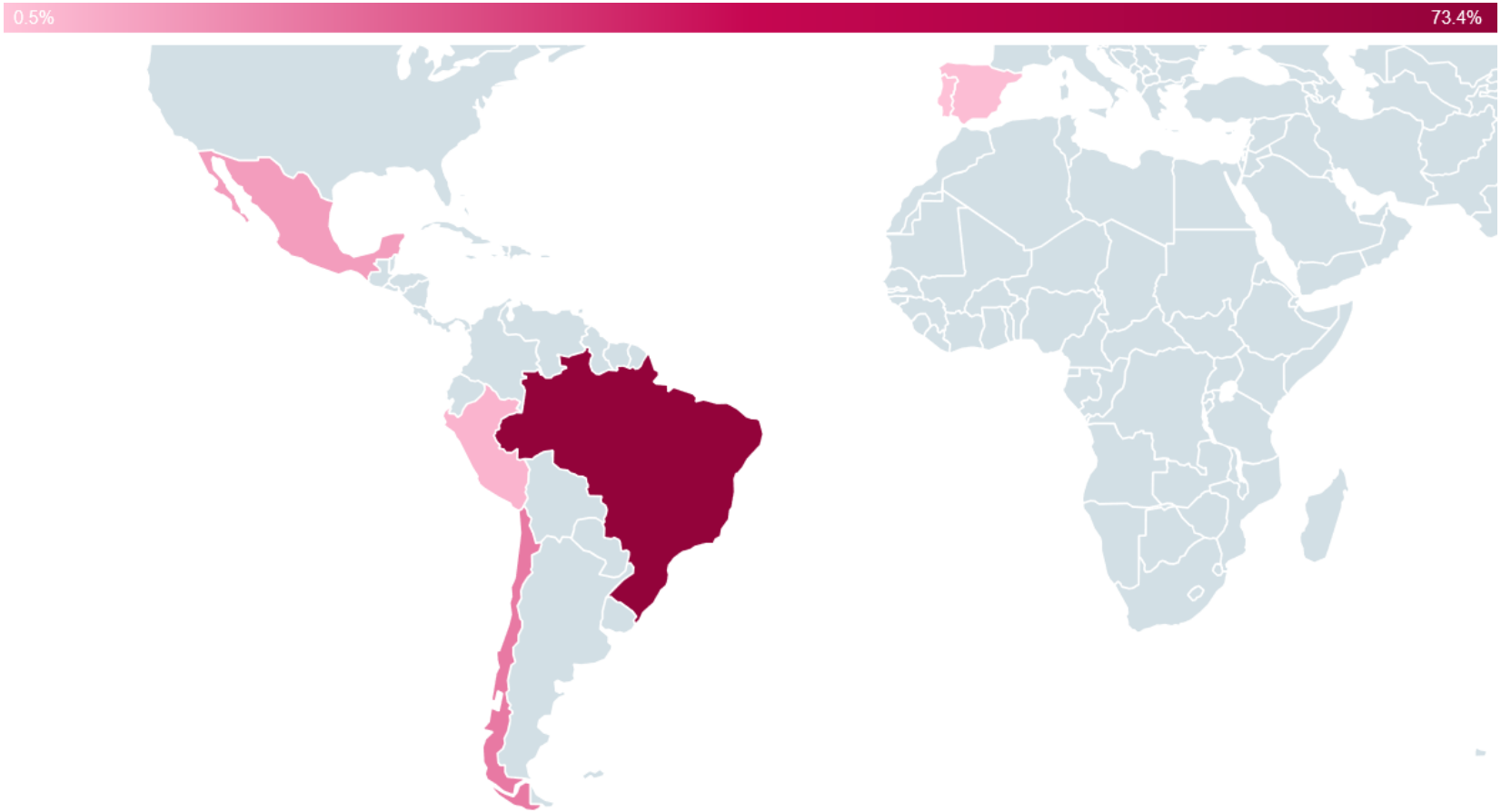
Mispadu (before Feb 2020)



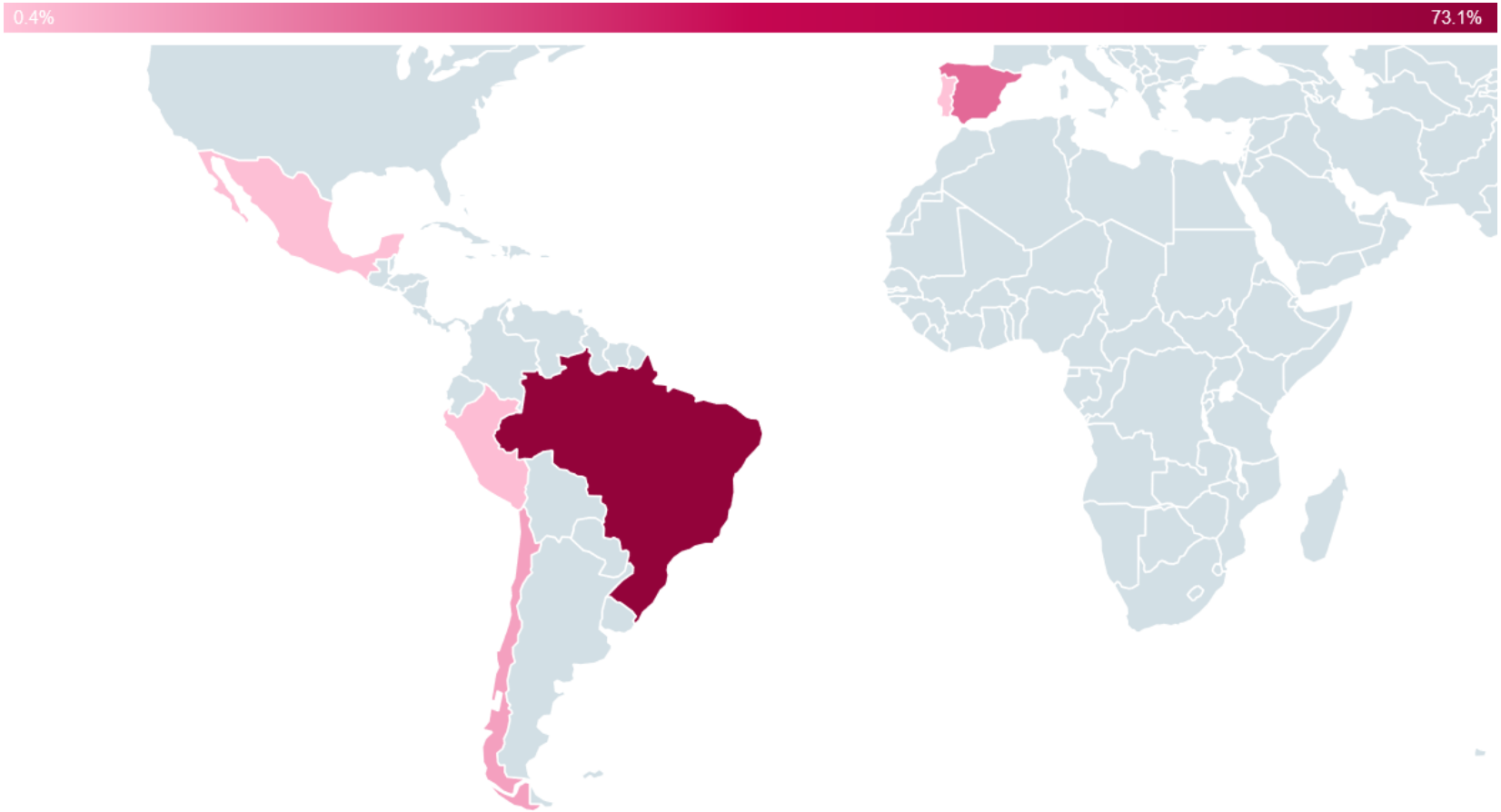
Mispadu (since Feb 2020)



Mekotio (before Mar 2020)

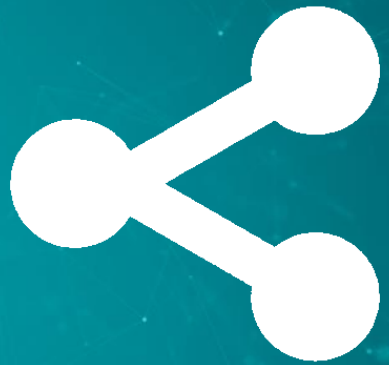


Mekotio (since Mar 2020)



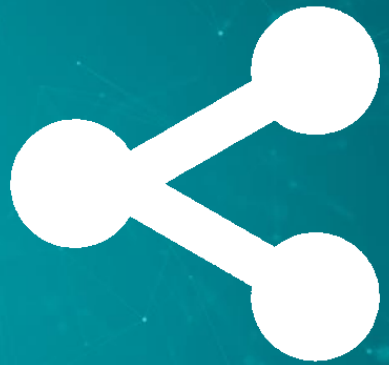
What is shared between LATAM banking trojans?

- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
  - Logic, specific chains, ZIP, MSI
- Execution methods
  - LoLBins, AU3 abusing, DLL side-loading, abused products
- Geographical distribution
  - South America, Mexico, Spain, Portugal



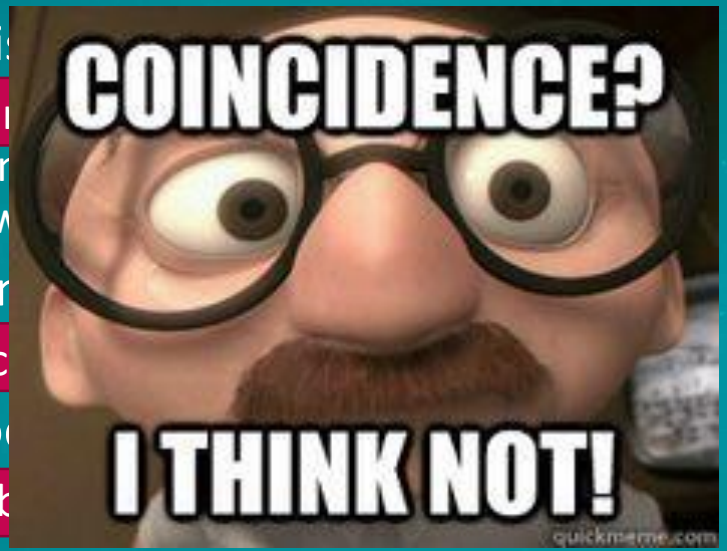
What is shared between LATAM banking trojans?

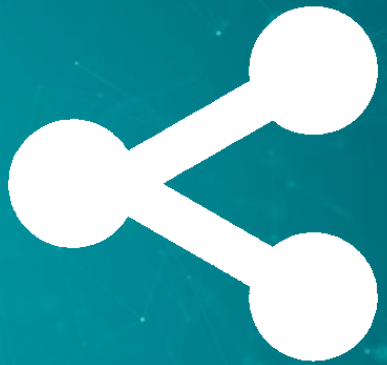
- Binary characteristics
  - Delphi, large binaries, logic, encryption schemes, implementation details, third-party components, anti-fraud software targeting
- Distribution chains
  - Logic, specific chains, ZIP, MSI
- Execution methods
  - LoLBins, AU3 abusing, DLL side-loading, abused products
- Geographical distribution
  - South America, Mexico, Spain, Portugal



What is shared between LATAM banking trojans?

- Binary characteri
  - Delphi, large bi
    implementatio
    anti-fraud softw
- Distribution chai
  - Logic, specific c
- Execution metho
  - LoLBins, AU3 ab
    products
- Geographical distribution
  - South America, Mexico, Spain, Portugal





So who is behind LATAM banking trojans?

- Independent threat actors with the same ideas?

So who is behind LATAM banking trojans?

- Independent threat actors with the same ideas?
  - Hardly

So who is behind LATAM banking trojans?

- Independent threat actors with the same ideas?
  - Hardly
- One group maintaining all the families?

So who is behind LATAM banking trojans?

- Independent threat actors with the same ideas?
  - Hardly
- One group maintaining all the families?
  - Unlikely… Why so many? Repeating mistakes, missing features, …

So who is behind LATAM banking trojans?

- Independent threat actors with the same ideas?
  - Hardly
- One group maintaining all the families?
  - Unlikely… Why so many? Repeating mistakes, missing features, …
- Multiple threat actors cooperating?

So who is behind LATAM banking trojans?

- Independent threat actors with the same ideas?
  - Hardly
- One group maintaining all the families?
  - Unlikely… Why so many? Repeating mistakes, missing features, …
- Multiple threat actors cooperating?
  - Yes!

Conclusion

- LATAM banking trojans
  - are region-specific families with the same goal
  - share
    - Techniques, ideas & tools
  - show collective and continuous development
- Something we have never seen before

ENJOY SAFER TECHNOLOGY™

Jakub Souček

Malware Analyst

jakub.soucek@eset.cz

Martin Jirkal

PRG Analyst Team Lead

jirkal@eset.cz

www.eset.com | www.welivesecurity.com | @ESETresearch