



ROCKET KITTEN: A CAMPAIGN WITH 9 LIVES

THREAT INTELLIGENCE AND RESEARCH



TABLE OF CONTENTS

EXECUTIVE SUMMARY	03
INVESTIGATION TIMELINE REVIEW.....	04
ROCKET KITTEN TOOLS & INFRASTRUCTURE	09
GEFILTE PHISH—BEST SERVED COLD	11
WOOLGERED—HOISTED BY THEIR OWN PETARD	18
REELED IN—PHISHING LOGS ANALYSIS	25
EPILOGUE	27
APPENDIX A—INDICATORS OF COMPROMISE.....	28
APPENDIX B—MPK TECHNICAL DESCRIPTION	33

EXECUTIVE SUMMARY

Since early 2014, an attacker group of Iranian origin has been actively targeting persons of interest by means of malware infection, supported by persistent spear phishing campaigns. This cyber-espionage group was dubbed 'Rocket Kitten,' and remains active as of this writing, with reported attacks as recent as October 2015.

The Rocket Kitten group and its attacks have been analyzed on numerous occasions by several vendors and security professionals, resulting in various reports describing the group's method of operation, tools and techniques.

Characterized by relatively unsophisticated technical merit and extensive use of spear phishing, the group targeted individuals and organizations in the Middle East (including targets inside Iran itself), as well as across Europe and in the United States. Many of these targets were successfully compromised by various pieces of custom-written malware; and despite identification and flagging of their infrastructure, the attackers have struck again-and-again by making minor changes to their tools or phishing domains.

Check Point has obtained a complete target listing from the attackers' servers; among confirmed victims are high ranking defense officials, embassies of various target countries, notable Iran researchers, human rights activists, media and journalists, academic institutions and various scholars, including scientists in the fields of physics and nuclear sciences.

This report provides a summary of the findings including:

- New evidence obtained during Check Point's independent investigation into attacker infrastructure, including previously unpublished malware indicators.
- Information that appears to reveal the full extent of operations over the past year, and provides unique insight into target profiles and attacker operation internals.
- Analysis of attack data to reveal details on victims and specific industries that may have special significance to Iranian political and military interests.
- Analysis of attacker mistakes that appear to reveal the true identity of the main developer behind the group's activities (a.k.a. "Wool3n.H4T"), detailed for the first time.

It is our hope this report and measures taken over the past few weeks lead to an effective shutdown of attacker operations (current generation of tools and infrastructure). While Check Point customers are protected against all known variants of this threat, we urge fellow security vendors and malware research professionals to extend malicious IoC (Indicators-of-Compromise) coverage in current protection infrastructure.

INVESTIGATION TIMELINE REVIEW

[If you are familiar with previous publications and interested in Check Point's new insights, you may skip this section.]

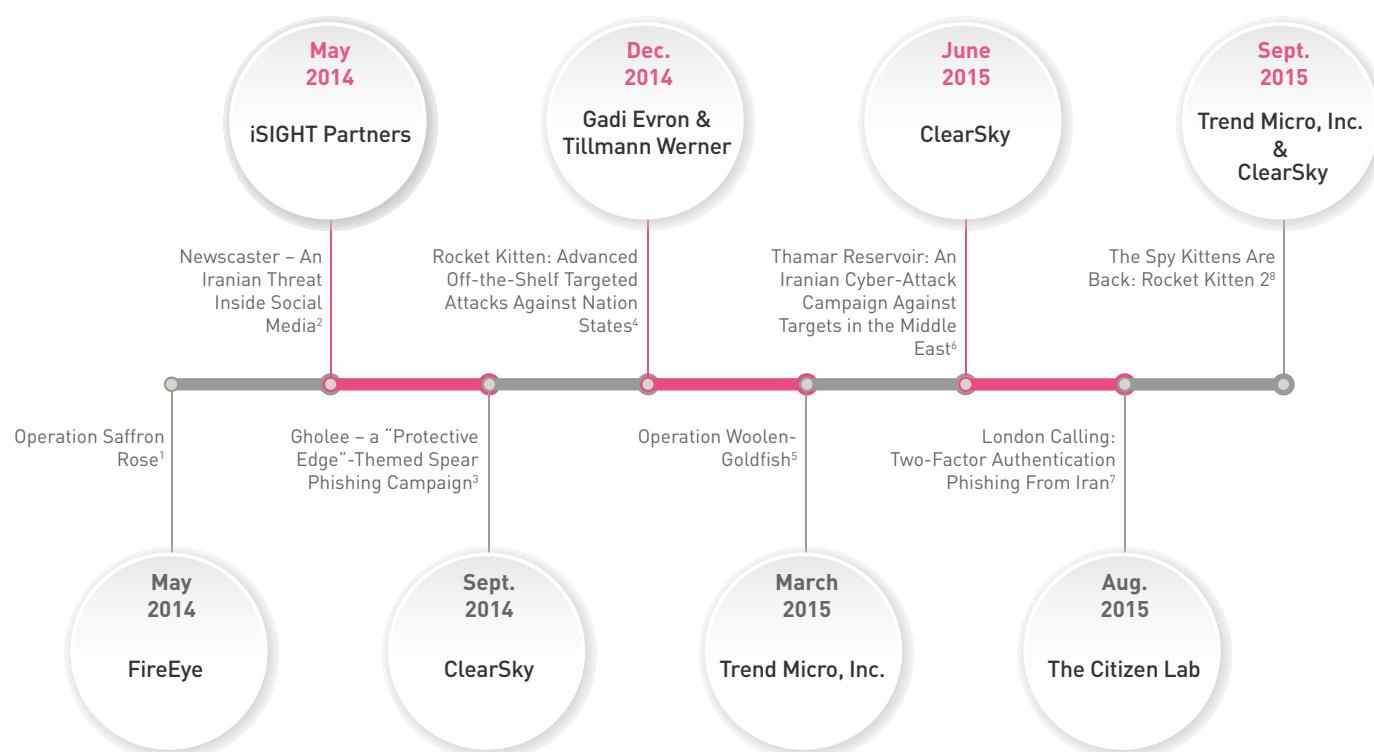
The Rocket Kitten campaign/actor group has been studied and analyzed on multiple occasions by different vendors, threat intelligence groups and individual researchers. In a repeating challenge in the malware research domain, we have seen different reports introduce a myriad of code names and operation names for what may very well be the same campaign/actors.

In contrast to malware naming schemes, all reports are in unanimous agreement with strong indications of the campaign's Iranian origin. This thesis is supported by the individuals and verticals targeted, as part of a plethora of circumstantial and direct evidence. While we should keep in mind digital evidence can be forged and tailored to falsely masquerade as any attacker to deceive a forensic analyst, the overwhelming amount of independent evidence collected over years of attack activity render the notion of a false campaign extremely improbable.

Despite all the reporting and sharing of malicious indicators, Check Point has detected continued active attacks using the same methods and infrastructure. These findings were confirmed by other security vendors, as well as Check Point's research partners.

It seems as if the attackers, unsophisticated as they are, are completely undeterred by the western security industry's revelations and publications. Often with the simple replacement of a domain name and minor updates to their malware tools, they continue to carry out their operation undisturbed.

Let us try to review and briefly summarize points of interest from the publications so far.



¹ <https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-operation-saffron-rose.pdf>

² <http://www.isightpartners.com/2014/05/newscaster-iranian-threat-inside-social-media/>

³ <http://www.clearskysec.com/gholee-a-protective-edge-themed-spear-phishing-campaign/>

⁴ <https://www.youtube.com/watch?v=WIhKovlHDJ0>

⁵ <https://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-operation-woolen-goldfish.pdf>

⁶ <http://www.clearskysec.com/wp-content/uploads/2015/06/Thamar-Reservoir-public1.pdf>

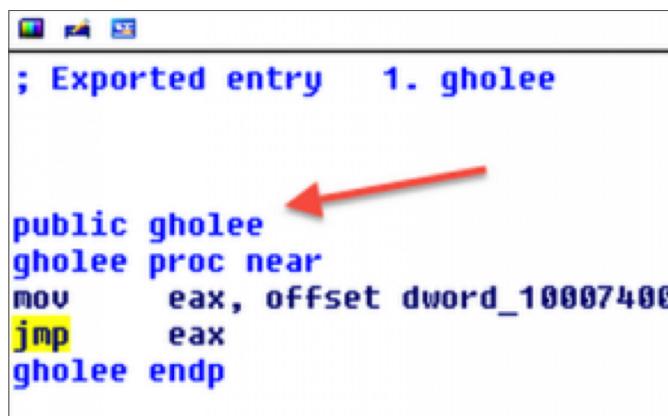
⁷ https://citizenlab.org/2015/08/iran_two_factor_phishing/

⁸ <http://documents.trendmicro.com/assets/wp/wp-the-spy-kittens-are-back.pdf>

The May 2014 ‘Operation Saffron Rose’ publication identifies an Iranian hacking group formerly named ‘Ajax Security’ (code-named ‘Flying Kitten’ by CrowdStrike) engaged in active spear phishing attacks on Iranian dissidents (those attempting to circumvent government traffic monitoring). This group is potentially linked to more recent Rocket Kitten attacks (different tools, yet very similar mode of operation and phishing domain naming scheme). No concrete evidence of such link has been presented yet.

Newscaster by iSight Partners was released the same month, to detail similar efforts of persistent spear phishing backed by false social media identities pertaining to be journalists of the fake news web site ‘newsonair.org.’ iSight, who reportedly cooperated with the FBI, provides a clear Iranian attribution to these efforts. The report specifies the attackers targeted policymakers, senior military personnel and defense industry organizations in the US, UK and Israel. We did not find direct evidence linking this activity to Rocket Kitten.

ClearSky’s September 2014 blog post first described active attacks using a piece of malware they dubbed ‘**Gholee**’ (as appears in a malicious payload export function, potentially named after a popular [Iranian singer](#)⁹). The researcher points to initial leads into other attacks and notes the threat is currently undetected by the overwhelming majority of AV products.



A screenshot of a debugger window showing assembly code. The title bar reads “; Exported entry 1. gholee”. The assembly code is:

```
public gholee
gholee proc near
mov    eax, offset dword_10007400
jmp    eax
gholee endp
```

A red arrow points to the label “public gholee”.

Image 1—the ‘gholee’ export name as noted by ClearSky.

Gadi & Tillman’s presentation at 31c3 (the 31st Chaos Communication Congress in Germany) was the first clear identification of the Rocket Kitten attacker group, continuing the CrowdStrike naming scheme for Iranian attacker groups. The publication introduced the involvement of hacker persona ‘**Wool3n.H4t**’ and other identities in forensic evidence obtained from the malicious documents.



A screenshot of a XML document showing core properties. The XML code is:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cp:coreProperties xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-properties" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:dcmitype="http://purl.org/dc/dcmitype/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<cp:coreProperties><dc:creator>
    Wool3n.H4t
</dc:creator>
<cp:lastModifiedBy>
    Wool3n.H4t
</cp:lastModifiedBy>
<dcterms:created xsi:type="dcterms:W3CDTF">
    2014-04-23T04:03:01Z
</dcterms:created>
<dcterms:modified xsi:type="dcterms:W3CDTF">
    2014-04-23T06:12:00Z
</dcterms:modified>
</cp:coreProperties>
```

Image 2—Forensic artifact in malicious document hinting to file creator as noted by Tillman Werner & Gadi Evron.

⁹ <https://www.youtube.com/watch?v=yNFA8l0kleQ>

The researchers followed to describe two pieces of malware used by the attackers:

- A deeper look into ClearSky's 'Ghlee' determined it is the 'wrapper' component of an off-the-shelf penetration testing tool originally authored by Argentina-based [Core Security](#). This legitimate PT tool, named 'Core Impact,' was illegally repurposed and used for malicious attacks by the Rocket Kitten group.
- A .NET-based credential stealer that pilfers known certain credential storage locations in the infected computer and e-mails them to 'wool3n.h4t@gmail.com'. This tool appears to be named 'FireMalv' by the attackers.

Trend Micro's March 2015 publication reintroduces the 'Ghlee' malware (as GHOLE) campaign, and describes 'Operation Woolen Goldfish,' as well as an additional 'CWoolger'—an unsophisticated key-logger apparently named 'woolger' (likely a Portmanteau for 'wool3n keylogger') written in C++, and present evidence showing its existence starting 2011.

```
C:\Users\Wool3n.H4t\Documents\Visual Studio 2010\Projects\C-CPP\CWoolger\Release\CWoolger.pdb
```

The researchers continue to point at the very likely attribution to the Wool3n.H4T identity as the malware author, whose only online reference was found in an Iranian blogging platform.



Image 4—wool3nh4t.blog.ir as pointed to by Trend Micro researchers

In this publication, Trend Micro researchers document Rocket Kitten's minor update to the Ghlee malware (the 'ghlee' function was renamed to 'function'), supposedly to avoid the Yara signature published by ClearSky and document the existence of Ghlee malware samples dating March 2011, as further evidence for historic attacker activity.

ClearSky continued their investigations into the group's activities and in June 2015 published a paper dubbing the attack campaign 'Thamar Reservoir' named after Dr. Thamar E. Gindin, herself a Rocket Kitten target. ClearSky researchers notably mention the breach of an Israeli academic institution to serve as a hosting service for the phishing web site, and follow to present an OPSEC (operational security) failure by the attackers that allowed ClearSky to learn of a detailed (partial) target list.

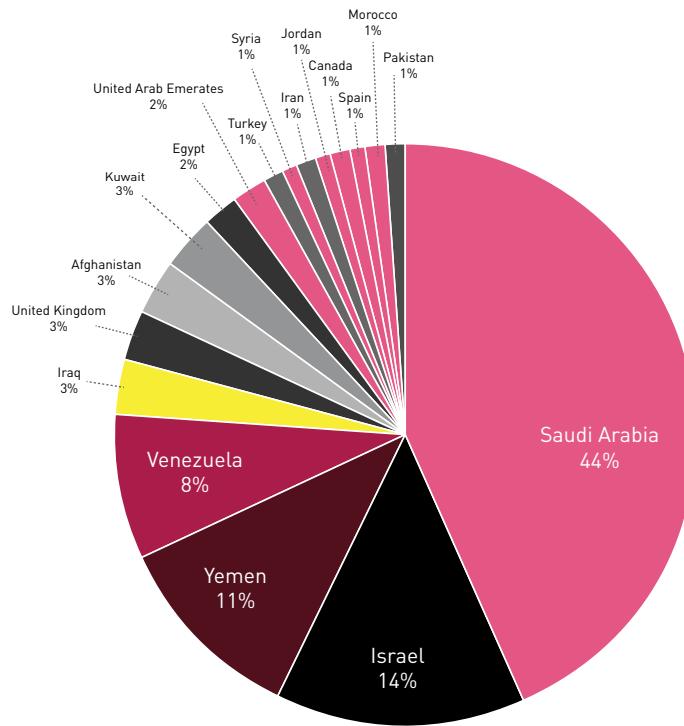


Image 5—Partial target country distribution as visible on the phishing server logs exposed by ClearSky

This list was analyzed to confirm a strong alignment with nation-state political interests, with specific victims known as adversarial or of intelligence value to Iran. ClearSky also reference an inadvertent public confirmation for the Iranian attribution by the US Department of Treasury in a memo which briefly appeared online, before being deleted.

ClearSky provided many examples of personalized phishing e-mails and communication, including phone calls to victims luring them to open these attachments, demonstrating the group's persistency and breadth of operations.

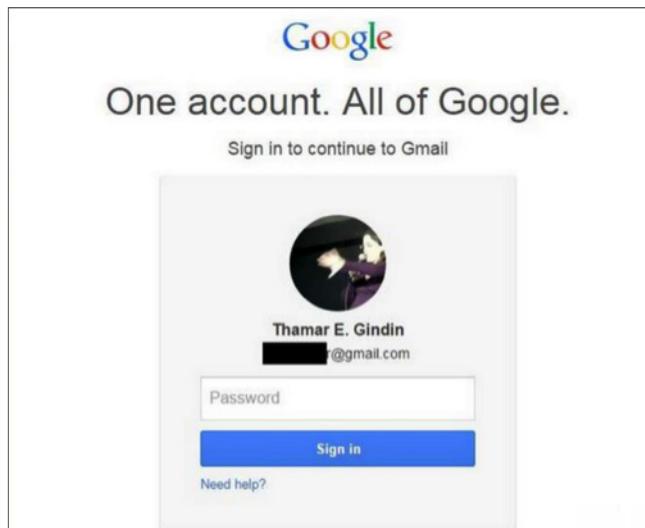


Image 6 – Custom- tailored phishing page as presented by ClearSky

The same phishing phone calling scheme was detailed in an August 2015 report by Citizen Lab, describing attempts to lure victims to provide their two-factor authentication tokens. In these attempts, victims receive tailored calls from a person who has clearly researched them, prompting them to take action on received e-mails. Among targeted victims Citizen Lab mention EFF's Director for International Freedom of Speech Jillian York. The Citizen Lab report describes overlapping phishing domains with ones previously reported, confirming a link with Rocket Kitten.

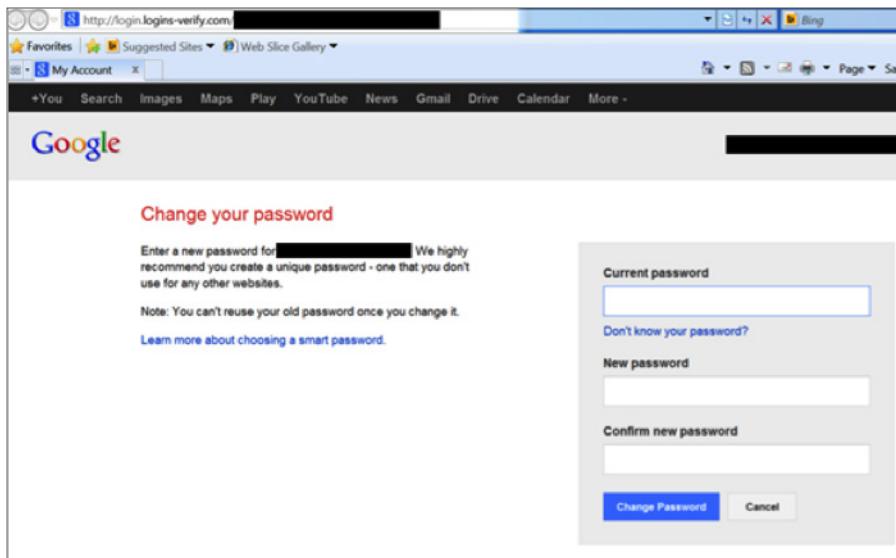


Image 7—Google password reset phishing page as presented by Citizen Lab

Interestingly, a special update to the Citizen Lab publication was added to include a response from a news outlet reported to be in close connection with Iranian intelligence, following allegations by exiled Iranian journalist Omid Memarian attributing these attacks to Iranian Revolutionary Guards¹⁰ with 'no doubt.' The response mocks the 'Western Media fishing in muddy waters' and describe the allegations as 'weird.'

The latest paper from Trend Micro and ClearSky (dated September 2015) goes a great length to detail the group's profile and mode of operations so far, and introduces a few more attack incidents, as well as a new 'downloader' piece of malware.

¹⁰ A branch of the Iranian armed forces, sworn to protect the country's Islamic system and prevent 'foreign interference'.

ROCKET KITTEN TOOLS & INFRASTRUCTURE

The Rocket Kitten attacker group's main attack vector is spear-phishing. An effective phishing campaign requires nothing more than a tailored phishing page, hosted on a cheaply-available web server. As described in previous publications, the Rocket Kitten attackers make extensive use of various phishing schemes, often including back-and-forth e-mail correspondence with the victims, or even phone calls to establish legitimacy and reason to open the malicious attachment.

Actual malicious attachments detected in this campaign varied between a set of custom-written malware pieces, or 'downloader' components that, in turn, fetch the malware from a remote server and execute it on the victim machine.

Additionally, we have witnessed many attacks using various 'web hacking' tools and suites, in attempt to break into victim web sites.

Previously reported custom-malware included:

- **CWoolger**—a C++ based 'woolen key-logger.' The malware records all keystrokes and sends out key-log data to a hard-coded FTP server.
- **Wrapper/Gholee**—repurposed Core Impact penetration testing tool. The malware allows a platform for remote access, pivoting for lateral movement and further malware installation.
- **FireMalv**—a .NET based Firefox credential stealer. This tool copies passwords stored in the Firefox browser storage.

Check Point investigations additionally discovered the attackers using the following:

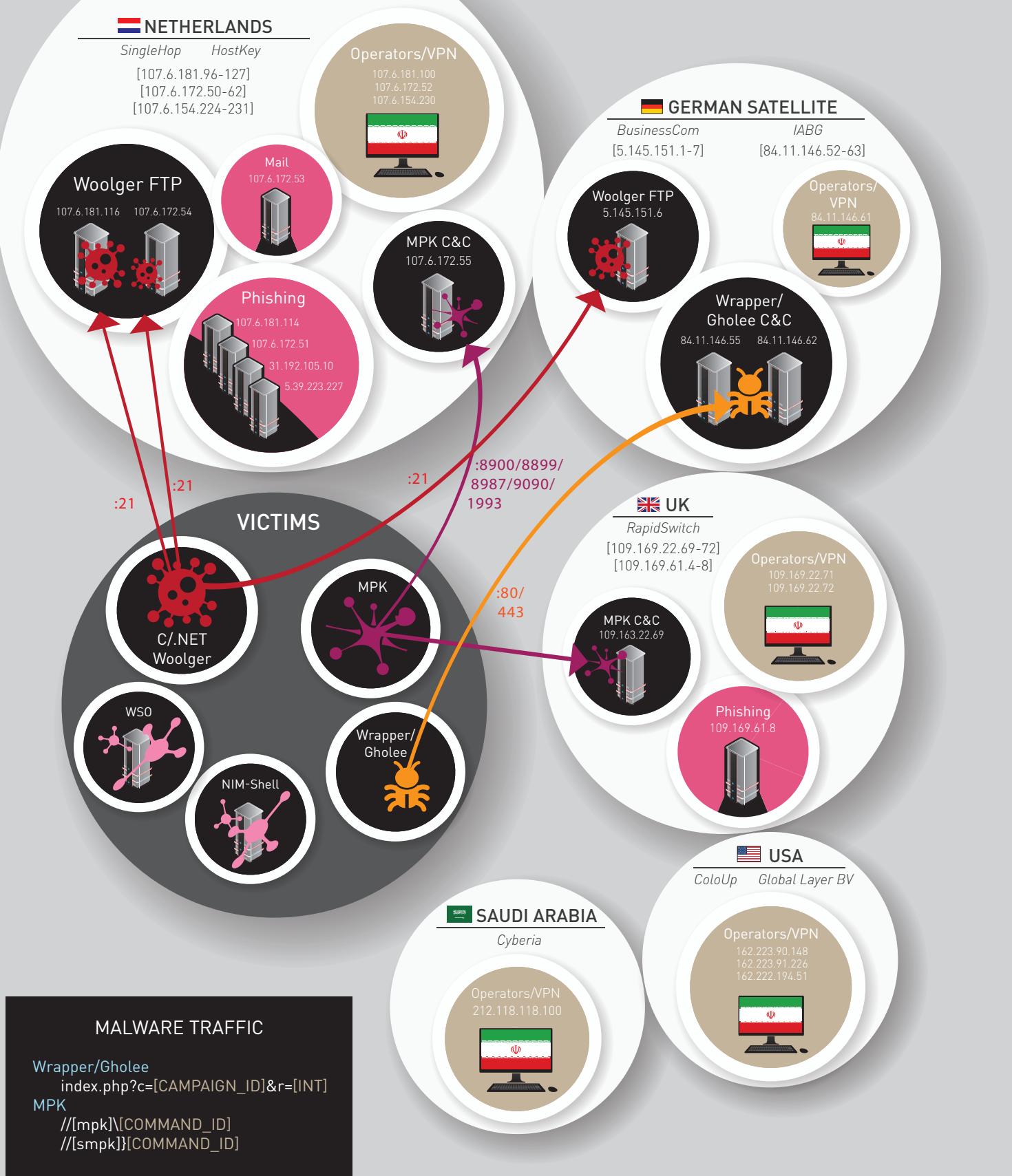
- **.NETWoolger**—a .NET based 'woolen key-logger.' This malware is functionally similar to CWoolger. The attackers seem to use them interchangeably, as alternate infection mechanisms (in case one is detected at the victim computer).
- **MPK**—a custom RAT of wider functionality. The malware allows key-logging, as well as remote command execution, screenshot grabbing and traffic monitoring. For a detailed technical description of the MPK malware see Appendix B.

In addition to custom-written malware, we have seen the attackers use various hacking and scanning tools to attack victim web-sites.

- **Metasploit**—An open-source, extensible penetration testing platform. Metasploit's 'meterpreter' payload was wrapped in an executable file and distributed as a RAT attached to phishing emails by the attackers.
- **Havij & SQLMap**—SQL injection tools; Havij originates in Iranian development, while SQLMap is an open source project.
- **Acunetix & Netsparker**—off-the-shelf web vulnerability scanners, attempting to automatically discover and exploit vulnerabilities in common web platforms.
- **WSO Web Shell**—a well-known web shell - PHP script that allows backdoor access on a hacked server. Typically deployed after successful compromise to allow further actions.
- **NIM-Shell**—a web shell of Iranian hacker group origin with similar functionality. Additionally uses Perl scripts on the hacked server.

Web hacking attempts were detected to originate from various IP ranges, occasionally immediately adjacent to known Rocket Kitten C&C servers. We can estimate the attack operators either used these servers directly, or configured them as Proxy/VPN endpoints to channel their attacks.

Combining the research work done so far with observed attacks by Check Point, we can map out a diagram overviewing the attacker's infrastructure.



- We have no reason to believe any of the mentioned providers are related to the malicious activity. The campaign operators likely masqueraded as a legitimate customer or hacked into the servers without the knowledge of the service provider.
- Specified ranges are likely to be assigned in whole for the attackers use. Due to the dynamic nature of IP assignment, these may expire after the release of this report.
- Because of the way satellite communications work, the infrastructure geo-located to Germany may not be physically located in that country. It would be an educated speculation to assume the servers are physically located in Iran. This assumption is supported by several indicators, including registrant details.

GEFILTE PHISH—BEST SERVED COLD

After learning of an active attack incident from the Rocket Kitten group on a customer network, Check Point researchers decided to actively join the investigation. While the recent paper from Trend Micro and ClearSky ('The Spy Kittens Are Back: Rocket Kitten 2') does extensively cover the campaign's narrative, we aimed to seek confirmation that our analyzed attack was positively connected to the same campaign and set out to provide additional value and insight.

Upon learning of the attack, we attempted to communicate with the phishing web server and gather primary reconnaissance. We learned the same IP address was used for multiple malicious domains. Noting the server on this IP address was alive and well, we decided to probe and question that particular server's purpose.

What we found took us all by surprise.

We started our web probe by making scripted GET requests attempting to browse to well-known paths. A minute later, we were excited to find a 200 OK response for a few requests, including `/xampp` and none other than `/phpmyadmin`(!).

Suspecting false positive results by our scripts, we typed in `/xampp` into our browser and watched with awe:



Image 8—A default configuration of XAMPP—on a live attacker server!

We curiously entered the direct path into our web browser and loaded the `phpmyadmin` interface.

It wasn't until we actually submitted a query on the server, when we understood that `phpmyadmin` had been configured to **allow password-less root access** to any browsing visitor.

"Such a gaping hole must be a decoy" we immediately thought. There is no way nation-state attackers would err in such amateur fashion, leaving their phishing server database exposed... would they?

If only they had paid attention to the ‘XAMPP Security’ page:

Subject	Status
These XAMPP pages are accessible by network for everyone Every XAMPP demo page you are right now looking at is accessible for everyone over network. Everyone who knows your IP address can see these pages.	UNSECURE
The MySQL admin user root has NO password Every local user on Windows box can access your MySQL database with administrator rights. You should set a password.	UNSECURE
PhpMyAdmin is free accessible by network PhpMyAdmin is accessible by network without password. The configuration 'httpd' or 'cookie' in the 'config.inc.php' can help.	UNKNOWN

These XAMPP pages are accessible by network for everyone
Every XAMPP demo page you are right now looking at is accessible for everyone over network. UNSECURE

Please consider this: With more XAMPP security some examples will NOT execute error free. If you use PHP in "safe mode" for example some functions of this security frontend will not working anymore. Often even more security means less functionality at the same time.

The XAMPP default ports:
http 21/tcp # File Transfer [Control] (XAMPP: FTP Default Port)

Image 9 – “The MySQL admin user root has NO password—UNSECURE”

Happily browsing through the free-for-all exposed database, we quickly noted numerous schemas; most of them were completely empty (for testing purposes?), with one specific schema standing out: ‘**phakeddb**’.

Image 10—phakeddb schema – note utf8_persian_ci collation for several tables

‘phakeddb’ contained a set of very interesting tables and data sets; the kind of data sets that fuel the fantasies of malware campaign researchers. Browsing these tables, we found the *phishing web application*, likely to be a custom development by the Rocket Kitten attackers. The web application would, upon operator instruction, generate the target-specific personalized phishing page for the targeted service (Gmail, YouTube, Hotmail, etc...).

As we later learn, this platform was named ‘Oyun Management System’ by the attackers.

Let us first look at the ‘users’ table:

user_id	user_name	user_pass	user_nickname	user_created_date	user_lock	isadmin	pcount
49	admin	e10adc3949ba59abbe56e057f20f883e	super admin	2014-08-09	0	IUM RT P MT UPL	22
50	anonymous	09d2b6cc9114ed718d9145fa40ed04f8	Anonymous	2014-08-09	0		8
51	merah	c12c563b4584be86a94dcba01aa80d0a	mire	2014-08-17	0		11
52	124	e87aa71e42c7de4780f448c8e92b50cd	razavi	2014-08-17	0	0	18
53	kaveh	46ec41ac0432182829250c0da50f89bb	kaveh	2014-08-17	0	0	10
54	ahzab	7a96925c26ec83a134f2014b77e01211	Ahzab	2014-08-20	0		40
55	attache	827ccb0eea8a706c4c34a16891f84e7b	irakli	2014-08-20	0	0	35
59	amirhosein	e89b359ba9008c1e1fda2bbe3374893e	ParsAAA	2014-08-21	0	0	10
60	john	e10adc3949ba59abbe56e057f20f883e	john	2014-08-24	0		10

Image 11—the ‘users’ table

The attackers log in to the application, just like any other web platform, in order to set up their phishing campaigns. This server seems to have been deployed August 2014, when all users were created.

And the hash type of passwd fields used? You may not be surprised to learn they used **unsalted MD5 hashes**.

That’s actually not the most oblivious malpractice in this system, however; the hash for the user named “super admin” (assigned with all possible permissions) is e10adc3949ba59abbe56e057f20f883e. Hobbyist cryptographers may recognize this string as the MD5 hash for “123456”.

Looking at user names, we can spot some potentially Persian names or aliases such as merah, kaveh, ahzab or amirhosein. These were potentially the campaign ‘operators’—tasked with social engineering and tailoring a phishing page per target. (hint: “123456” was not the only trivially crackable password in this list)

Moving on to the intriguing “conversation” table, this appears to be an experimental messaging feature between attackers. Unfortunately, it was rarely used.

msg_id	sntr_id	date	content	viewed
17	51	2014-09-16 03:00:00	http://syntaxmarketing.com.au/wp-content/uploads/2...	1
18	51	2014-09-17 00:00:00	https://www.youtube.com/watch?v=VZmdhw3axw	1
22	54	2014-09-22 00:00:00	http://profiles.google.com.inc.gs/?_schema=1326&m...	0
24	52	2014-10-01 12:10:25	https://mail.mail2.mod.gov.af/owa/auth/logon.aspx?...	0
25	52	2014-10-05 00:00:00	https://cid-c4351db11d15e77f.users.storage.live.co...	0
26	52	2014-10-05 00:00:00	10/r	0
28	51	2014-10-26 00:00:00	https://accounts.google.com/VA?c=COm3Jn_2-CSogEQ9...	0
29	51	2014-10-29 00:00:00	Adrese asli: http://outlook.com/owa/biu.ac.il redi...	0
30	51	2014-10-29 00:00:00	http://www.youtube.com/watch?feature=youtu.be&v=S...	0
31	51	2014-10-29 00:00:00	http://www.youtube.com/watch?feature=youtu.be&v=S...	0
32	51	2014-10-29 00:00:00	Sign in to continue to YouTube	0
33	55	2014-10-29 00:00:00	pleaze 20 subject for me. tank you attache	0
34	51	2014-11-24 00:00:00	http://profiles.facebook.in/loginuser/?_schema=198...	0

Image 12—the ‘conversation’ table

Most messages include links to various phishing domain pages, perfectly correlating to reported attack pages, proving this database is indeed in direct correlation to the attacks.

Interestingly enough, we can see user id 55 (correlating with the 'attache' username in the users table) making a request:

please 20 subject for me.
tank you
attache

Later on, user id 60 ('john') pleads:

As 'boos' is Farsi for 'kiss', 'bos bos' might be considered the Persian version of 'xoxo'.

What does that mean? What projects and subjects are this system's users referring to?

A glimpse into ‘requiretypes’ lays out a much clearer description for this system’s purpose:

req_id	req_exp	req_desc	req_exam
18	%VEA%	Victime Email Address	wool3n.h4t@gmail.com
19	%VFN%	Victim Full Name	John Kerry
20	%VAU%	Victim Avatar Url	http://exam.com/avatar.jpg
21	%DESC_1%	Description (1)	No Example
22	%DESC_2%	Description (2)	No Example
25	%VF%	Victim Family	Kerry
26	%VN%	Victim Name	John
27	%FPLT%	Fake Page Link Text (For email message)	No Example
28	%FPLU%	Fake Page Link Url	No Example
29	%VUN%	Victim User Name	wool3n.h4t
30	%EMAIL_1%	Email Address (1)	No Example
31	%EMAIL_2%	Email Address (2)	No Example
32	%VNN%	Victim Nick Name	Woolen
33	%IMG_1%	Image Url (1)	No Example
34	%IMG_2%	Image Url (2)	No Example
35	%SFE%	Supervisor Email (Fake Mail)	supervisor@ybsoft.com
36	%TMP%	Temp Value	No Example

Image 13—the ‘requiretypes’ table

We can see template codes for phishing pages, including the descriptive “Victim Full Name,” “Victim User Name” field values. It seems that this application generates the custom phishing templates using these custom fields. Even more telling, we have examples for each field, reintroducing us to our Wool3n.H4T friend (author of key-logger malware by the same group), repeatedly mentioned in this column. This introduces the reasonable possibility that Wool3n.H4T himself wrote this “phishing application” as a supporting tool for the campaign.

There is an intriguing ‘supervisor@ybsoft.com’ reference, too, but ybsoft.com is currently registered to a Chinese electronics shop, so no luck in that direction.

The real jackpot, however, is still ahead.

When we opened the ‘projects’ table, we momentarily lost our breath. A ‘project’ is apparently a single victim (target e-mail address), assigned with a ‘proj_id’, a tasked operator, and the specific link that was generated to be sent to this victim. We just hit 1842 records including all victims attacked starting August 2014 and all the way to August 2015 (when this database was accessed).

proj_id	proj_name	page_id	user_id	proj_date	proj_url	time_stamp
1148	[REDACTED]@gmail.com	41	54	2014-09-14	http://google-profiles.com/?_schema=1148&rnd=32331	1432651592
1149	[REDACTED].com	41	54	2014-09-14	http://google-profiles.com/?_schema=1149&rnd=24266	
1151	[REDACTED]ns@hotmail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1151&...	1410751030
1152	[REDACTED]ail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1152&...	1410804467
1153	[REDACTED]ail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1153&...	
1154	[REDACTED]otmail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1154&...	
1155	[REDACTED]@gmail.com	41	54	2014-09-14	http://google-profiles.com/?_schema=1155&rnd=31367	1410812432
1156	[REDACTED]@gmail.com	41	54	2014-09-14	http://google-profiles.com/?_schema=1156&rnd=17253	1411011128
1157	[REDACTED]@hotmail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1157&...	
1158	[REDACTED]ail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1158&...	
1159	[REDACTED]ail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1159&...	
1160	[REDACTED]gmail.com	41	54	2014-09-14	http://google-profiles.com/?_schema=1160&rnd=10193	1410810084
1161	[REDACTED]il.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1161&...	
1162	[REDACTED]mail.com	48	50	2014-09-14	http://outlook.profile.hmail.us/new/?_schema=1162&...	1432652579
1163	[REDACTED]	51	50	2014-09-15	[REDACTED]	1414391670
1164	[REDACTED]GMAIL.COM	41	54	2014-09-15	http://google-profiles.com/?_schema=1164&rnd=20601	1411341118
1165	[REDACTED]ail.com	41	54	2014-09-15	http://google-profiles.com/?_schema=1165&rnd=28997	1410814674
1166	[REDACTED]mail.com	41	54	2014-09-15	http://google-profiles.com/?_schema=1166&rnd=32180	1432650666
1169	[REDACTED]	42	49	2014-09-15	http://members.google-it.info/?_schema=1169&rnd=24...	1432650666
1170	[REDACTED]n@gmail.com	41	54	2014-09-15	http://members.google-it.info/?_schema=1170&rnd=26...	1432650680
1171	[REDACTED].com	41	54	2014-09-15	http://members.google-it.info/?_schema=1171&rnd=17...	1432650680

Image 14—the ‘projects’ table

Not only do we have the e-mail addresses of all victims, we also have the template values for their respective phishing pages (in the ‘projectmailrequirevalues’ table)!. For example a ‘Google Sign-In’ page normally displays the full name of the victim, as well as a public avatar defined by the user. The attackers had to replicate this look and feel, and filled the database with full names, addresses and photos for every targeted victim.

We verified and retrieved names and images of previously reported victims as expected.

proj_id	req_id	req_value
397	18	[REDACTED]
397	19	Hatoon [REDACTED]
397	20	https://lh3.googleusercontent.com/-FGF5dJEFXxE/AAA...
398	18	[REDACTED]@gmail.com
398	19	Tariq [REDACTED]
398	20	https://lh3.googleusercontent.com/uFp_tsTJboUY7kue...
400	18	[REDACTED]@gmail.com
400	19	Kazem [REDACTED]
400	20	https://lh3.googleusercontent.com/-XdUlqdMkCWA/AAA...
401	18	[REDACTED]@gmail.com
401	19	harry [REDACTED]
401	20	https://lh3.googleusercontent.com/-cOgYe1Js6yw/AAA...
402	18	[REDACTED]@gmail.com
402	19	Adel [REDACTED]
402	20	https://lh3.googleusercontent.com/uFp_tsTJboUY7kue...
403	18	[REDACTED]@gmail.com
403	19	nouf [REDACTED]
403	20	https://lh5.googleusercontent.com/-kdaLUY728Hk/AAA...
406	18	hussain [REDACTED]@gmail.com
406	19	Hussain [REDACTED]
406	20	https://lh6.googleusercontent.com/J_O9TZsZJOA/AAA...

Image 15 – the ‘projectmailrequirevalues’ table

But what does ‘projectlogs’ contain?

Is that what we think it is?

log_id	proj_id	log	date	viewed
50971	2583	</br>Page viewed on Monday 2015-08-17 At 12:19:45<... 2015-08-17	1	
50972	2583	</br>Page viewed on Monday 2015-08-17 At 12:21:50<... 2015-08-17	1	
50973	2583	</br>Page viewed on Monday 2015-08-17 At 12:23:42<... 2015-08-17	1	
50974	2583	</br>Page viewed on Monday 2015-08-17 At 12:23:56<... 2015-08-17	1	
50976	2583	</br>Page viewed on Monday 2015-08-17 At 12:27:35<... 2015-08-17	1	
50993	2585	</br>Page viewed on Monday 2015-08-17 At 14:11:0<... 2015-08-17	1	
50994	2585	</br>Page viewed on Monday 2015-08-17 At 14:11:1<... 2015-08-17	1	
50995	2586	</br>Page viewed on Monday 2015-08-17 At 14:17:43<... 2015-08-17	1	
50996	2586	</br>Page viewed on Monday 2015-08-17 At 14:17:58<... 2015-08-17	1	
50997	2586	</br>Page viewed on Monday 2015-08-17 At 14:18:11<... 2015-08-17	1	
51000	2586	</br>Page viewed on Monday 2015-08-17 At 14:49:6<... 2015-08-17	1	
51001	2586	</br>Page viewed on Monday 2015-08-17 At 14:51:44<... 2015-08-17	1	
51002	2586	</br>Page viewed on Monday 2015-08-17 At 14:51:49<... 2015-08-17	1	
51003	2586	</br>Page viewed on Monday 2015-08-17 At 14:52:38<... 2015-08-17	1	
51106	2573	</br>Page viewed on Tuesday 2015-08-18 At 9:4:23<... 2015-08-17	0	
51107	2588	</br>Page viewed on Tuesday 2015-08-18 At 9:38:39<... 2015-08-17	1	
51108	2588	</br>Data sent from victim: submitted = 1... 2015-08-17	1	
51109	2588	</br>Page viewed on Tuesday 2015-08-18 At 9:39:36<... 2015-08-17	1	
51110	2588	</br>Page viewed on Tuesday 2015-08-18 At 9:39:54<... 2015-08-17	1	
51111	2588	</br>Page viewed on Tuesday 2015-08-18 At 9:41:22<... 2015-08-17	1	
51112	2588	</br>Data sent from victim: submitted = 1... 2015-08-17	1	
51135	2579	</br>Page viewed on Tuesday 2015-08-18 At 14:4:50<... 2015-08-18	0	

Image 16 —log of every access to any phishing page on that server

This table contained a log entry for every access to any phishing page, including the credentials provided by the victims, if successfully fooled. We can now use this data to gather insightful analytics on spear phishing activity over one year spanning August 2014 to August 2015. Please see the attack log analysis section in this report.

Continuing our server probe, we discovered a similarly exposed ‘Webalizer’ interface, providing useful analytics including counters and frequently accessed links.

Monthly Statistics for August 2015		
Total Hits	745288	
Total Files	730870	
Total Pages	736350	
Total Visits	31815	
Total KBytes	31251982	
Total Unique Sites	40	
Total Unique URLs	351	
Total Unique Referrers	226	
Total Unique Usernames	1	
Total Unique User Agents	76	
	Avg	Max
Hits per Hour	2070	47561
Hits per Day	49685	487457
Files per Day	48724	473402
Pages per Day	49090	478545
Sites per Day	2	14
Visits per Day	2121	19986
KBytes per Day	2083465	20193386
Hits by Response Code		
Code 200 - OK	98.07%	730870
Code 206 - Partial Content	0.00%	1
Code 301 - Moved Permanently	0.01%	82
Code 302 - Found	0.16%	1218
Code 303 - See Other	0.01%	60
Code 304 - Not Modified	0.00%	32
Code 400 - Bad Request	0.01%	44
Code 401 - Unauthorized	0.47%	3493
Code 403 - Forbidden	0.01%	41
Code 404 - Not Found	1.14%	8464
Code 406 - Not Acceptable	0.13%	966
Code 408 - Request Timeout	0.00%	1
Code 413 - Request Entity Too Large	0.00%	1

Image 17—Webscraper statistics for August 2015

The Webscraper interface neatly presented us with a lot of useful metadata, including “Top 40 visitor IPs”—clearly identifying attacker access to the site, and providing us with many leads for the remainder of the investigation. Interestingly enough, we also found some referrer headers, leading to a path on the same server:



Username :

Password :



Image 18—login screen

In what can be described as a ‘hacker secret access’ portal—we seem to have reached the web interface of the phishing platform. Testing the ‘admin’ credentials we previously “cracked”—we get:



Image 19—the “Oyun Mangement System (OMS)” [sic]

We now learn the attackers name this system “Oyun” and actually used Larry Page’s public profile picture as admin’s avatar. The remainder of the interface simply allows access to the phakeddb database, including insertion, editing of “projects” (/targets) and even the internal chat platform as evident in “conversations.”

WOOLGERED—HOISTED BY THEIR OWN PETARD

Using credentials hard-coded into the woolen key getLogger, we were able to retrieve numerous woolger DAT files (key-logs), as uploaded from victims around the world.

As apparent, the same hard-coded FTP credentials were, in fact, Administrator credentials on the C&C Windows server itself, which had the C\$ and D\$ NetBIOS/SMB administrative shares openly accessible from the WAN.

Name	Date modified
FTP Contents	09/07/2014 14:45
inetpub	09/07/2014 14:37
PerfLogs	17/02/2015 15:15
Program Files	09/07/2014 14:37
Program Files (x86)	16/09/2015 07:35
ProgramData	12/02/2015 06:36
Users	07/02/2015 13:51
wamp	25/02/2015 22:54
Windows	10/08/2015 14:37

*Image 20—if you didn’t want to allow researchers to have administrator access to your C&C server
<captain_hindsight.png>
you shouldn’t have hard-coded administrator credentials into your malware.*

Among many logged keystroke files containing stolen data, we stumbled on an astonishing discovery: the Rocket Kitten attackers had, in fact, **infected their own workstations**, apparently as ‘test-runs’ for woolger. The attackers failed to remove these files from the C&C server, demonstrating, yet again, utter lack of OPSEC.

Of peculiar interest to us were logs made by none other than 'Wool3n.H4t' himself:

```
3 -----
4 C:\Users\Wool3n.H4t\Desktop\
5 -----
6 [Enter]
7 -----
8 -----
9 *C:\Users\Wool3n.H4t\Desktop\
10 -----
11 this[Enter] [Alt]this is a test
```

Image 21—The test worked.

Would you be surprised what followed in the same log file?

```
33
34 CWoolger (Running) - Microsoft Visual Studio
35 -----
36 [Control] PTX
37
38 -----
39 postgresql for windows 7 64 bit - Google Search - Mozilla Firefox
40 -----
41 [Control] SYN Unicode [Bottom] [Bottom] [Enter]
42
43 -----
44 CWoolger (Running) - Microsoft Visual Studio
45
46 W([Del] [Right] [Right] [Right] [Right] [Right] [Right] [Right] [Right])
47
48 -----
49 *C:\Users\Wool3n.H4t\Desktop\
50
51 [Enter] [Enter] [Enter] hgi jjg jbjhg jhgjhgu k jhk
52
53 -----
54 C:\Users\Wool3n.H4t\AppData\Local\Temp\wlg.dat - Notepad++
55 -----
56 y
```

Image 22—The attacker testing his own tool.

Yes, we actually just witnessed Wool3n.H4T switching through his open windows, including a Microsoft Visual Studio live debugging session of a project called... 'CWoolger'

In another log, we observe the specific programming of loading of the 'wsc.vbs' script, as observed in the Trend Micro publication and others. At this point, there should be no question—we are looking at the malware author's development workstation.

```
22 -----
23 Google Translate - Mozilla Firefox (English (United States))
24 -----
25 mutex
26
27
28 -----
29 CWolger - Microsoft Visual Studio (English (United States))
30 -----
31 [Control] DC3DC3[Control] DC3DC3DC3[Top][Enter]char script_file_path =
[Back][Back][Back][255];[Enter]strc[Control] p[Enter](scr[Control] i[Enter],temp_[Control]
);[Enter]strc[Control] [Enter](scr[Control] ,""[Left][Esc]wsc.vbs[Right]);scr[Control]
[Control] DC3DC3[Control] DC3DC3DC3[Enter][Enter][Top][Top]syste[Control]
[Enter](scr[Control] );[Control] DC3
32
```

Image 23—Mutexes and thread-safety should be the least of your worries.

The next log shows us the attacker wanted to test whether his tool would accurately capture credentials entered into a Firefox HTTP authentication prompt, and thus he entered his own C&C server...

```
2 -----
3 new 1 - Notepad++ (English (United States))
4 -----
5 s
6 -----
7 *new 1 - Notepad++ (English (United States))
8 -----
9 alarm
10 -----
11 Untitled - Notepad (Persian)
12 -----
13 o,fd ucdcl ;phdd[Back] [Back] [Back] [Back]p[Back] [hdd ?
14 -----
15 New Tab - Mozilla Firefox (Persian)
16 -----
17 [Alt]ftp[Bottom] [Enter]
18 -----
19 Authentication Required (English (United States))
20 -----
21 administrator[Tab] [Enter]
22 -----
23 Problem loading page - Mozilla Firefox (English (United States))
24 -----
25 [Enter]
26 -----
27 Authentication Required (English (United States))
28 -----
29 administrator[Tab] [Enter]
30 -----
31 Index of ftp://107.6.181.116/woolen/ - Mozilla Firefox (English (United States))
32 -----
33 [Del] [Enter]
34 -----
35 Index of ftp://107.6.181.116/ - Mozilla Firefox (English (United States))
36 -----
```

Image 24

All of Wool3n.H4T's retrieved logs were dated October 2014.

Then, we spotted this log segment:

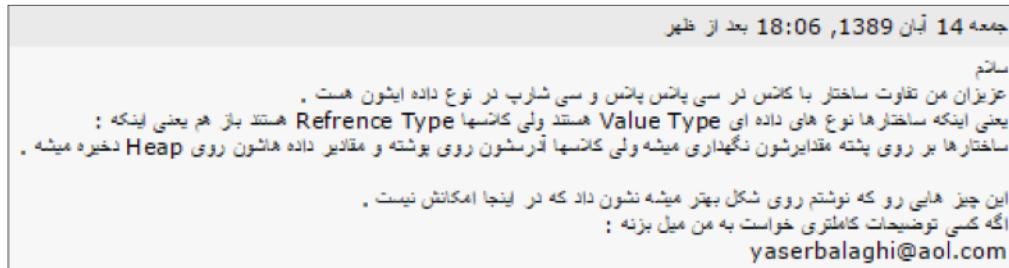
```
10 -----
11 New Tab - Mozilla Firefox (English (United States))
12 -----
13 mail.aol.com[Enter]
14 -----
15 -----
16 -----
17 AOL Mail: Simple, Free, Fun - Mozilla Firefox (English (United States))
18 -----
19 yaserbalaghi[Tab]123456756[Enter]
```

Image 25 - 'AOL Mail' already narrowed it

Recorded under the Wool3n.H4T name, a user logs into AOL mail with username 'yaserbalaghi.'

Could it be the same 'Yaser' as noted in the recent Trend Micro and ClearSky paper? ('D:\Yaser Logers\CWoolger')... Could it explain the Phakeddb reference to "ybsoft"? We don't know at this point; we have to go deeper.

'yaserbalaghi@aol.com' appears to give a technical answer in a long C++ thread in an Iranian programmers forum ("Barname Nevis") in Iranian 'Solar Hijri' calendar year 1389 (2010-2011):



The same yaserbalaghi user made several posts, also linking to various programming instructional videos in the subjects of ASP.NET and AJAX, jQuery and SQL injections as instructed by him using screen capture software.

Careful watching of the videos allowed us to learn a few interesting details. For starters, Yaser Balaghi is a Microsoft Visual Studio 2010 user, with familiarity of several tools observed to be used during the 'Rocket Kitten' campaign.

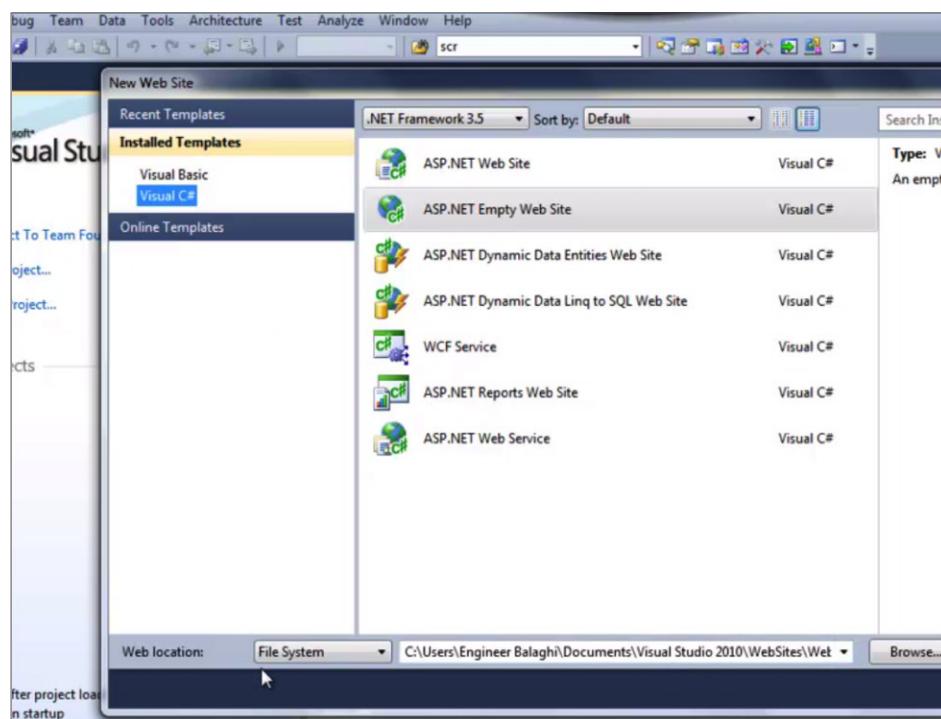


Image 26—Screenshot taken from instructional video by Yaser Balaghi (Engineer Balaghi)

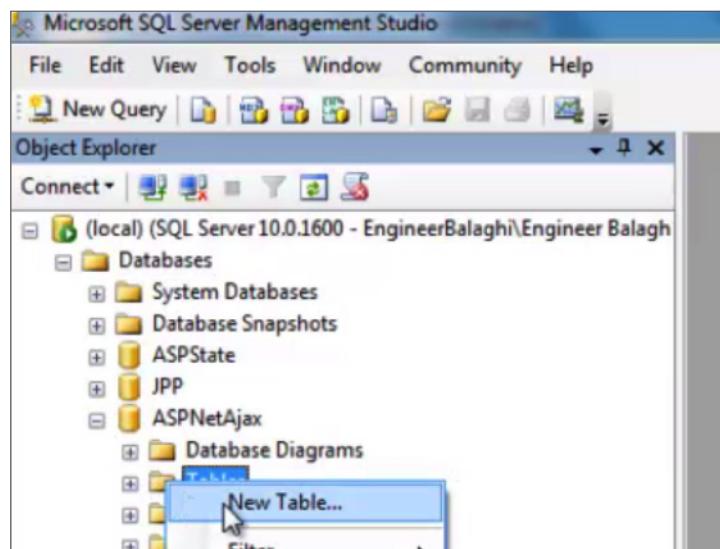


Image 27—EngineerBalaghi host name

Further inspecting the user names and host names evident in the screen captures, we noticed we were actually in possession of logged keystrokes from an “infected computer” where the user name was “Engineer Balaghi,” strengthening our suspicions. However, we can’t be sure yet; Yaser Balaghi may be a common name or perhaps this is someone related to Wool3n.H4T or the attackers.

A few minutes later, and we spotted this gem of an OPSEC mistake in the SQLi instructional video, which precisely provided the smoking gun we were after:

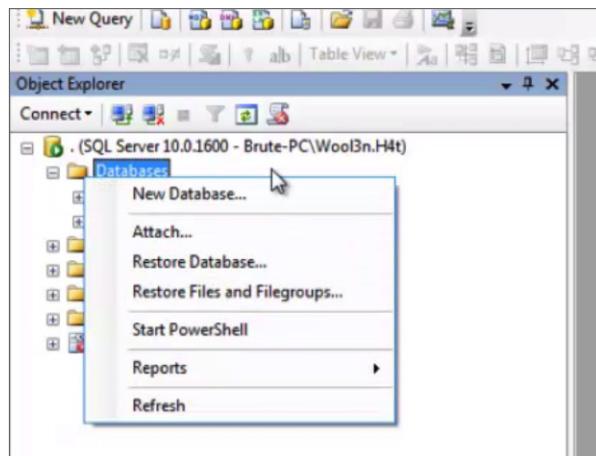
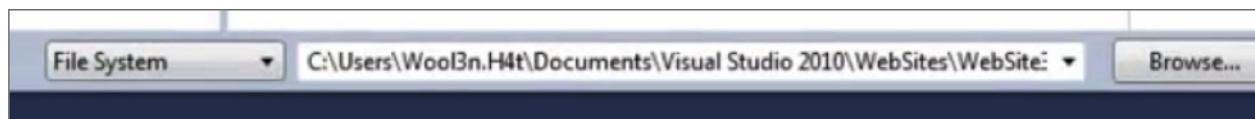


Image 28—Watching an hour of Farsi SQL injection tutorial has its rewards



Wool3n.H4t is caught red-handed. One of his many mistakes, he was now caught giving a public tutorial while logged in under his secret alias, otherwise unlinked with his real identity. These videos were recorded February 2014, prior to the first clear Rocket Kitten attack wave mid-year.

A quick glance on WOOL3n.Hat's desktop reveals a striking match with web hacking attack tools previously described in Rocket Kitten's arsenal.



Image 29—Havij, Acunetix, Netsparker, SQLMap, wamp, and oh—is that IDA properly licensed?

A few online queries later, we are getting numerous results, cross-referenced to verify as the same Yaser Balaghi, now the main suspect to hold the Wool3n.H4T identity.

Engineer Yaser Balaghi is not only an active member of various programming forums—he had a web site (www.eng-balaghi.com, gone offline since August 2014, still available in the Wayback Machine). In the available archived version of the site he described himself as a “programmer, analyst, consultant and lecturer,” and made himself available for hire.

A screenshot of a Stack Overflow user profile for 'Yaser Balaghi'. The profile includes a headshot of a man with dark hair and a suit, the name 'Yaser Balaghi' in bold, and a bio stating 'Software Engineer 1. Windows Developer 2. Web Developer 3. Android Developer 4. Web Security'. Below the bio, it says '1 REPUTATION'. Navigation tabs at the top include 'Profile' (which is active) and 'Activity'. Other tabs like 'Questions', 'Tags', and 'Users' are also visible.

Image 30—Yaser Balaghi's stackoverflow account

If all that wasn't enough, we also managed to retrieve an updated resume for Tehran-based Engineer Balaghi:



Image 31 – Yaser Balaghi's Resume (2013)

Islamic Azad University Computer Software Graduate Balaghi lists his job experience, including “Technical Director and Team Leader of Software Development Team (**Private**)” (highlighted in original), as well as “Head of Security and Hacking (legal and ethical) (**Private**)”. Later, he goes as far as listing sample accomplishments and completed projects, including the development and system design for a “Phishing Attacks System” ordered by “a cyber-organization”.

- ✓ طراحی نرم افزار Brute Force به سفارش یک ارگان سایبری (**Private**)
 - ✓ طراحی سامانه حلات فیشنینگ به سفارش یک ارگان سایبری (**Private**)
 - ✓ طراحی نرم افزار File Binder به سفارش یک ارگان سایبری (**Private**)
 - ✓ طراحی یک بد افزار ویندوزی با زبان پایتون به سفارش یک ارگان سایبری (**Private**)
 - ✓ انجام دهها پروژه سایت هکینگ به سفارش یک ارگان سایبری (**Private**)
 - ✓ طراحی و پیاده سازی بسیاری از پروژه های نرم افزاری و همچنین Hack Tools ها و کار های متفرقه .
- ✓ Designing Brute Force Software ordered by a cyber-organization (**private**)
 - ✓ Designing phishing attacks system ordered by a cyber-organization (**private**)
 - ✓ Designing File Binder Software ordered by a cyber-organization (**private**)
 - ✓ Designing a windows Malware in *Python language* ordered by a cyber-organization (**private**)
 - ✓ Accomplishing tens of hacking projects ordered by a cyber-organization (**private**)
 - ✓ Designing and executing a lot of software projects and also hack tools and miscellaneous projects

Image 32 (original and translation)—we kid you not.

We could go on, but the main lesson of this section can be: if you don't want people to know you created malware for the government, don't list it in your CV.

REELED IN—PHISHING LOGS ANALYSIS

As reported so far, the attackers persistently e-mailed, called and responded with fake identities, tailored for each victim. The attackers clearly read the public reports about them, respond and adapt their tactics, occasionally showing a creative mindset.

In one reported case, the attacked posed under the true identity of a ClearSky researcher, referencing the recent Rocket Kitten report, attaching “detection software” that does exactly the opposite. This an interesting tactic, worthy of mentioning in social engineering classes. It would be wise at this point to mention that the release of report does not include any accompanying detection or protection tools other than the existing Check Point software blades. If you received this report with an attached executable, it is likely a malicious lure.

In another case, the attackers sent a malicious attachment using the identity of a previously known targeted victim. The Israeli recipient of that attachment was wary enough to suspect the origin of the e-mail and responded with a query: “Is that you or are the Iranians in your computer again?” To which the attackers responded (in perfect non-Google-Translate Hebrew): “The Iranians will never return to my computer!”

That very well may have been the talk of the day at the Tehran operations center, possibly featured in an email printout in the main dining room.

As the Rocket Kitten group’s behavior was well characterized in previous publications (see the recent report from Trend Micro and ClearSky). We will focus on new insights based on our analysis of the ‘Oyun’ system victim database. We understand that this database contains a partial view, starting August 2014 to August 2015. While the data can be successfully correlated with logs collected from other servers, we have no visibility of e-mails with malicious attachments (as opposed to phishing links to steal credentials), or any complete web hacking log for attacker activity.

The sheer volume of the target database suggests an extensive operation, the work of a group of people over months. The logs included the visiting IP address geo-located country. Our analysis shows the following distribution:

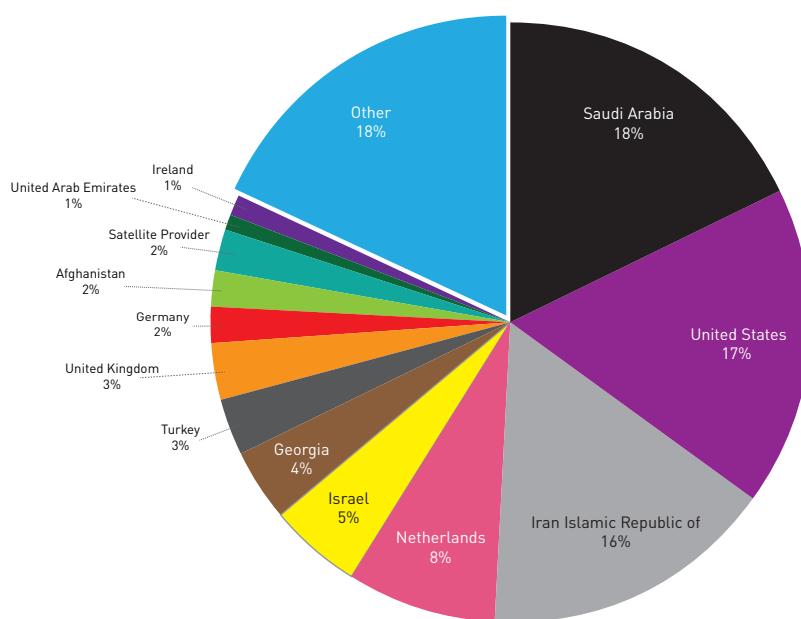


Chart 1—Phishing visitors’ country distribution

We have studied the visitor data to determine this includes many attacker accesses to test the site functionality. We know the attackers used addresses from Iran, as well as VPN access from the US, Germany, Saudi Arabia and the Netherlands. The data must be interpreted taking these facts into consideration.

Our primary filtering dismissed around 25% of logs and 15% of projects as 'test runs' for the system. The following is based on the remaining seemingly valid entries.

Charting the phishing logs over time, we can observe the following timeline:

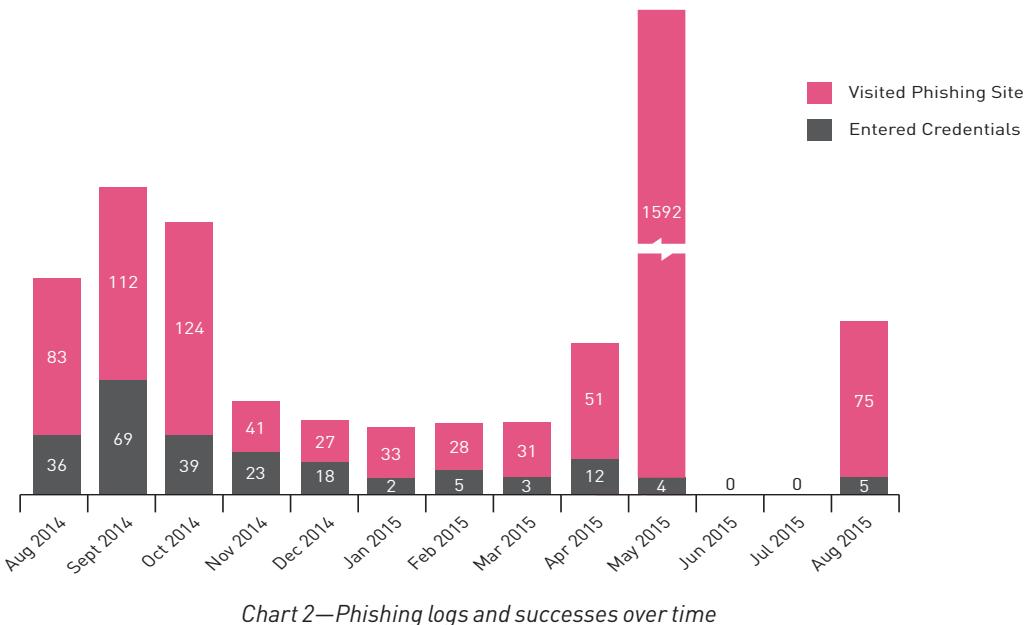


Chart 2—Phishing logs and successes over time

We can study this data to make a few interesting observations:

- On average, all phishing pages on this server had 26% success in fooling victims to enter their credentials. These are surprisingly high results, potentially attributed to persistency and well-targeted e-mails.
- On May 26, 2015, there is a unique peak of access to the site, with minimal successes. When analyzed, these accesses appear mostly in 3 batches over periods of minutes, with incremental 'project_ids' and no data provided, from Israeli IP addresses. We can safely discard these as researcher probes, attempting to 'brute-force' phishing pages, immediately preceding the ClearSky June publication.
- The attackers seem to have shut down their platform on June and July (likely due to the publication) and resumed operations during August. We found evidence to suggest the database had been migrated from a previous server.

Slicing the projects table by user_id allows a unique internal look on operator assignment; while our target analysis is far from conclusive, we can share a reserved primary assessment of what each user was tasked with:

User	Projects	Target Profile
admin	83 projects	strictly system testing
anonymous	522 projects	this is one of the prominent users in the system, tasked with all around mixed targets—focused on Saudi Arabia, many human rights activists, CEOs and ministry officials.
merah	147 projects	assigned with all Israeli targets, notably including known physics and nuclear scientists, former military officials, national security and foreign policy researchers. This operator is probably a fluent Hebrew speaker.
kaveh	57 projects	very little activity—mostly testing and some Venezuelan targets.
ahzab	691 projects	one of the two busiest operators, in 2014 he targeted a vast amount of Saudi scholars and persons of influence, and later listed education and media outlets in Saudi Arabia.
124 & attache	233 projects	these users both showed clear targeting of defense sector victims, as well as embassies of Iran's neighboring countries and others. Quite fitting with one of their usernames, they listed several military attachés in their victim list. Notable targets include representatives from the United Arab Emirates, NATO and other regional posts in Afghanistan as well as Thailand and Turkey.
john	108 projects	During late 2014 he was directly tasked with Venezuela trade and finance targets, later in 2015 he moved to former Iranians living abroad—listing professors, scientists, journalists and investors

Despite our limited visibility, we can confirm many of these attacks were successful—the attackers gained confidential information from various targets all around the world.

EPILOGUE

We believe the Rocket Kitten case is an interesting case study for the malware research industry, exemplifying a continuing trend in the nation-state attacker profile we have witnessed over the past two years; cyber-espionage is no-longer reserved to organizations with monstrous budgets to hire thousands of cyber-warriors, operate password-cracking super-computer clusters or advanced research to infect your hard-drive firmware. Adversaries will often find simpler ways for effective compromise, such as creative phishing and simple custom malware.

In this case, as in other previously reported cases, it can be assumed that an official body recruited local hackers and diverted them from defacing web sites to targeted espionage at the service of their country. As is often the case with such inexperienced personnel, their limited training reflects in lack of operational security awareness, leaving a myriad of traces to the origin of the attack and their true identities (e.g. Yaser Balaghi, Mehdi Mahdavi and others).

Despite publications, code names and articles in security outlets - the same known attacker group continues to attack with minimal interruption.

Highlighting a repeating industry problem, minimal changes to existing malware often evade most current protection solutions. Effectively stopping attackers must involve action on top of analysis efforts.

We approached and will continue to approach hosting providers through the assistance of CERT coordination and other bodies in various countries. We hope these efforts are fruitful, and can help disable or reduce the attacking infrastructure.

If you would like to share important information regarding this campaign, please use icanhazrocket@checkpoint.com

APPENDIX A—INDICATORS OF COMPROMISE

Samples

All hashes are MD5 or SHA1

Lure Documents / Doppers

```
01c9cebbc39e273ac1f5af8b629a7327
08273c8a873c5925ae1563543af3715c
1685ba9dbdb0e136d68e0b1a80a969b5
177ef7faab3688572403730171ffb9c4
1ceca1757cb652ba7e5b0d45f2038955
266cf755a0a66776df9fd8cd2fee1f1
271a5f526a638a9ae712e6a5a64f3106
2cb23916ca60a63a67d974f4ddeb2a11
393bd2fd420eecf2d4ca9d61df75ff0c
395461588e273fab5734db56fa18051b
48573a150562c57742230583456b4c02
4bf2218eb068385ca1bfff8d609c0104
50d3f1708293f40a2c0c1f151c2c426f
54ee31eb1eed79d4ddffd1423d5f5e28
55ff220e38556ff902528ac984fc72dc
5a009a0d0c5ecaac1407fb32ee1c8172
5af0cbc18c6f8ed4fd1a3f68961f5452
60f5bc820cf38e78b51e1e20fed290b5
61a808ce0b645c4824d79865be8888ed
85b79953bf2b33fb6118dc04e4c30910
8ed01ac79680d84c0ee7a5f027d8b86a
9fc345c25e6ab94bca2db6ee95d2c861
ac94ee83c91ca784a88ff26cf85e273a
aeb9d12ecbe73bfa91616ebacf24831b
c9ea312c35e9ac0809f1c76044929f2f
d0c3f4c9896d41a7c42737134ffb4c2e
d14b3e0b82e3b5d6b9cc69b098f8126d
e1a5b4ffc612270425d5d31f4c336aa9
f68a0a3784a7edfc60ad9333ec209cbf
f8547010eb4238f8fb76f4e8a756e36d
0482fc2e332918456b9c97d8a9590781095b2b53
0f4bf1d89d080ed318597754e6d3930f8eec49b0
1a999a131144afe8cb7316ebb842da4f38101ac5
2627cdc3324375e6f41f93597a352573e45c0f1e
2c3edde41e9386bafef248b71974659543a3d774
46a995df8d9918ca0793404110904479b6adcb9f
4711f063a0c67fb11c05efdb40424377799efafdf
476489f75fed479f19bac02c79ce1befc62a6633
64ba130e627dd85c85d6534e769d239080e068dd
6571f2b9a0aea89f45899b256458da78ac51e6bb
788d881f3bb2c82e685a98d8f405f375c0ac2162
9579e65e3ae6f03ff7d362be05f9beca07a8b1b3
a9245de692c16f90747388c09e9d02c3ee34577e
ad6c9b003285e01fc6a02148917e95c780c7d751
ae18bb317909e16f765ba2e88c3d72d648db2798
b67572a18282e79974dc61ffb8ca3d0f4fc1b0
c485b0d59b28d37a1ac80380b0d7774bdb9d8248
c727b8c43943986a888a0428ae7161ff001bf603
```

```
e2728cabb35c210599e248d0da9791991e38eb41  
e6964d467bd99e20bfef556d4ad663934407fd7b  
ec692cf82aef16cf61574b5d15e5c5f8135df288  
ed5615ffb5578f1addee66f571ec65a992c033a50  
f51de6c25ff8e1d9783ed5ac13a53d1c0ea3ef33  
f7f69c5ed94a03f6d57e9af33c2627ff69205f2
```

Wrapper / Gh0st

```
05523761ca296ec09afdf79477e5f18d  
08e424ac42e6efa361eccfdf3c13b21  
0b67eb08f09c0584b92f4e94ced778  
13039118daadbe87e337310403e64454  
14f2e86f11114c083856c92095d79256  
1b02ac8c0e1102faaee70f4026cad291  
223feb91efbe265696f318fb7c89c3fd  
3dd221b0ea6f863e086868b246a6a104  
4215d029dd26c29ce3e0cab530979b19  
48573a150562c57742230583456b4c02  
4b0edcd1d2953c26b6fc4298e8bf9150  
4cdc28ab6e426dc630638488743accfb  
58bcfe673d21634616d898c3127bd1bc  
60f5bc820cf38e78b51e1e20fed290b5  
63558e2980d1c6aaaf34beefb657866fe  
8a45dfec98dd96c86d933d9c1d6ef296  
8bd58db9c29c53197dd5d5f09704296e  
916be1b609ed3dc80e5039a1d8102e82  
a42cea20439789bd1d9a51d9063ae3e4  
b7de8927998f3604762096125e114042  
b884f67c247d3dd6c559372a8a31a898  
b8fb83d76eb67cbeed0b54c02a68256b  
c222199c9a7eb0d162d5e96955739447  
d5517542b5f8dc2010933ee17a846569  
da976a502a3afc4ba63611d47c625738  
ee41e7c97f417b07177ea420afe510a1  
f3c3ed556072209b60c3342ddefba0f9  
f89a4d4ae5cca6d69a5256c96111e707  
02b04563ef430797051aa13e48971d3490c80636  
07a77f8b9f0fcc93504dfba2d7d9d26246e5878f  
0b0cdf47363fd27bccbfba6d47b842e44a365723  
0b880fb3414374dbbf582217ee0288a76c904e9b  
22f6a61aa2d490b6a3bc36e93240d05b1e9b956a  
25d3688763e33eac1428622411d6dda1ec13dd43  
37ad0e426f4c423385f1609561422a947a956398  
476489f75fed479f19bac02c79ce1befc62a6633  
47b1c9caabe3ae681934a33cd6f3a1b311fd7f9f  
53340f9a49bc21a9e7267173566f4640376147d9  
58045d7a565f174df8efc0de98d6882675fbb07f  
62172eee1a4591bde2658175dd5b8652d5aead2a  
6e30d3ef2cd0856ff28adce4cc012853840f6440  
729f9ce76f20822f48dac827c37024fe4ab8ff70  
7ad0eb113bc575363a058f4bf21dbab8c8f7073a  
7fef48e1303e40110798dfec929ad88f1ad4fb08
```

```
8074ed48b99968f5d36a494cdeb9f80685beb0f5
86222ef166474e53f1eb6d7e6701713834e6fee7
c1edf6e3a271cf06030cc46cbd90074488c05564
c6db3e7e723f20ed3bcf4c53fc4748e9591f4c40
cabdf7e9920aeaa5eaca7f5415d97f564cdec11
ce03790d1df81165d092e89a077c495b75a14013
e6964d467bd99e20bfef556d4ad663934407fd7b
e8dbcde49c7f760165ebb0cb3452e4f1c24981f5
efd1c6a926095d36108177045db9ad21df926a6e
fa5b587ceb5d17f26fe580aca6c02ff2e20ad3c4
fd8793ce4ca23988562794b098b9ed20754f8a90
fe3436294f302a93fbac389291dd20b41b038cba
ffead364ae7a692afec91740d24649396e0fa981
```

FireMalv Credential Stealer

```
0b0e2c4789b895e8ac44b6ada284aec1
29d93b156bcfbcecf79c5ba389094796a1ba76ee
```

Woolen-Keylogger

```
0a22232c1d5add9d7aabdf630b6ed5af
0e2dc1cb6bda45d68ee9c751e37df73b
1a2b18cb40d82dc279eb2ef923c3abd0
1f7688653c272d5205f9070c2541a68c
3c6c1722acfb70bfa4453b69e99c98bb
662d094799e9c7108f35c00eb894205f
b4790618672197cab31681994bbc10a4
c72dce99e892bbf2537f5285a01985c0
f7e093d721d2616ecb9067934a615f70
f898eef9dfa04820bb2f798e063645a7
f9b235067b1c607b5b26896d465b6665
29968b0c4157f226761073333ff2e82b588ddf8e
5d334e0cb4ff58859e91f9e7f1c451ffdc7544c3
8e1bd64acd8bbe819ac60650eb1fa4f501d330ec
a42f1ad2360833baedd2d5f59354c4fc3820c475
a65b39d3919f15649106a039469013479a31ba4b
b9842058c88170cc45183aaaae4206c74e6c7351
c8096078f0f6c3fbb6d82c5b00211802168f9cba
d5b2b30fe2d4759c199e3659d561a50f88a7fb2e
db2b8f49b4e76c2f538a3a6b222c35547c802cef
eeb67e663b2fa980c6b228fc2e04304c8992401d
faf0fe422259d36494a0b2c9ccefe40dee978f31
```

MPK

```
014bf8a588f614883d3d8b96024cd278  
5c66b560f70c0b756bfc840b871864ce  
d1b526770abb441d771f4681872d2fcb  
eb6a21585899e702fc23b290d449af846123845f  
f2ed8cd0154ae4d6ecf52a0bcf5fa80c7095dcd2  
f710bd9ea40fd94c06d704c00e16a5941544378f
```

Network Traffic**Wrapper/Gholee**

HTTP/HTTPS [80/443]

index\ .php\ ?c=\w+\&r=\d+

Woolger

FTP [21] to 107.6.181.116, 107.6.172.54, 5.145.151.6

MPK

```
raw [8900,8899,8987,9090,1993] -  
\\\\[mpk\\]\\\d{4} example: //\[mpk]\2012  
\\\\[smpk\\}\\\d{4} example: //\[smpk]\1992
```

Domains

```
account.login.gfimail.us  
accounts.google.uk.to  
account-user.com  
drive-google.co  
drives-google.co  
gfimail.us  
gmail-member.us.to  
google-setting.com  
google-verify.com  
login.miicrosoftonline.us.to  
login.office365.uk.to  
logins-verify.com  
login-users.com  
mail.mail2.mod.gov.af.mail.al  
mail-verify.com  
my.idc.ac.il.my.to  
outlook.profile.com.hmail.us  
outlook.tau.ac.il.mail.al  
owa.inss.mises.org.il  
owas.haifa.ac.il.info.gf  
owas.haifa.us.to  
profile.gmail.us.to  
profile.google.uk.to  
profiles.facebook.in
```

profiles.google.com.inc.gs
profiles.googlemembers.com.home.kg
profiles-google.uk.to
qooqle.co
secure.www.cfr.us.to
service-logins.com
signin-users.com
signin-verify.com
signs-service.com
verification.google-it.info
video.qooqle.co
webmail.tau.ac.il.us.to
webmail.technion.ac.il.us.to
yahoo-profiles.uk.to
youtube.com.now.im

IP addresses

[107.6.181.96-127]

[107.6.172.50-62]

[107.6.154.224-231]

107.6.181.116

107.6.172.54

107.6.172.55

107.6.181.114

107.6.172.51

107.6.172.53

107.6.181.100

107.6.172.52

107.6.154.230

5.39.223.227

31.192.105.10

[5.145.151.1-7]

5.145.151.6

[84.11.146.52-63]

84.11.146.55

84.11.146.62

84.11.146.61

[109.169.22.69-72]

[109.169.61.4-8]

109.169.61.8

109.169.22.69

109.169.22.71

109.169.22.72

162.223.90.148

162.223.91.226

162.222.194.51

212.118.118.100

APPENDIX B—MPK TECHNICAL DESCRIPTION

The malware appears to be named 'MPK' by the attackers. This may be related to "Masoud_PK" as witnessed in the Iranian blogging web-site under the wool3n.h4t blog name.

Installation

For persistence, the malware will add itself to autorun under an "explorer" entry:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run  
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

The malware includes a Visual Basic script ('tmp.vbs') script, which will try to initially copy the malware executable to its destination:

```
Sub CopyFile(SourceFile, DestinationFile)  
    Set fso = CreateObject("Scripting.FileSystemObject")  
    Dim wasReadOnly  
    wasReadOnly = False  
    If fso.FileExists(DestinationFile) Then  
        If fso.GetFile(DestinationFile).Attributes And 1 Then  
            fso.GetFile(DestinationFile).Attributes = fso.GetFile(DestinationFile).Attributes - 1  
            wasReadOnly = True  
        End If  
        fso.DeleteFile DestinationFile, True  
    End If  
    fso.CopyFile SourceFile, DestinationFile, True  
    If wasReadOnly Then  
        fso.GetFile(DestinationFile).Attributes = fso.GetFile(DestinationFile).Attributes + 1  
    End If  
    Set fso = Nothing  
End Sub  
  
copyme = WScript.Arguments.Item(0)  
copyto = WScript.Arguments.Item(1)  
CopyFile copyme,copyto,0
```

Also, it will execute the following WScript, which will start the malware itself after exactly 9 seconds.

```
WScript.Sleep 9000  
CreateObject("WScript.Shell").Run "iexplorer.exe [1]"
```

Main operation

This malware is basically a RAT (Remote Access Trojan). It implements such functionality as a key-logger, sniffing TCP and UDP traffic, taking screenshots, as well as a remote command shell.

Also, it may gather a lot of information about the target system such as enumeration of files, drives, services, process information and the ability to send any file to the C&C server.

Less important, but still sensible information may be exfiltrated:

- Primary display resolution
- Has administrator rights or not
- Processor information
- Hostname information
- Windows version
- Service Pack version
- Amount of memory installed on the target system
- Network adapters and network configuration information
- TCP connections table

The following mutex will be created:

[2] opened

Then, the malware will check if the following mutex exists:

MyApp1.0

If it does, the malware will exit, so only one instance of the malware is allowed at a time. If not, it will continue to the main operation.

Keylogger

Keylogger stores keystrokes to the following file:

%TEMP%\log%d.txt

Here is a sample of malware key-log output:

```
(((((Hello new File)))))))
+++++
Window= VMware Accelerated AMD PCNet Adapter (Microsoft's Packet Scheduler) : Capturing -
Wireshark
+++++
[UP] [DOWN] [DOWN] [UP] [UP] [DOWN] [UP] [DOWN] [UP] [UP] [DOWN] [DOWN] [DOWN] [DOWN] [UP] [UP] [UP]
[DOWN] [DOWN] [DOWN] [DOWN] r
+++++
Window= Run
+++++
cmd [ENTER]

+++++
Window= C:\WINDOWS\system32\cmd.exe
+++++
notepad [ENTER]

+++++
Window= VMware Accelerated AMD PCNet Adapter (Microsoft's Packet Scheduler) : Capturing -
Wireshark
+++++
[DOWN] [UP]
+++++
Window= Untitled - Notepad
+++++
test test test
+++++
```

This file will be sent to remote C&C server later on.

If the malware detects an open "Gmail", "Yahoo" or "Outlook" window, it will add special processing so the attacker can easily recognize the data that is the most valuable to him. The following string will be appended to the output file:

```
\r//////////\r\nMail Find
```

Webcam capture

The malware may capture photos from an attached webcam. Files are first saved with the name test.bmp, later converted to JPEG and saved under the new file name Cam.jpg, eventually exfiltrated to the C&C server.

TCP Connection Table

The malware will gather available metadata regarding current TCP connections using the GetTcpTable API, and send a formatted version of the obtained data to the C&C server.

Screenshots

The malware may take screenshots. The filename used for screenshots is Screeny.jpeg.

Remote Shell (Live Command Execution)

The malware creates the following process as a live command prompt:

```
cmd.exe /c cmd.exe
```

This process' output and input are attached and redirected via pipes to the remote C&C server, allowing the operator to type in commands to control the victim computer. The following line is first sent to the server:

```
Welcome To mpkshell Command Line (This Message Send From Server)
```

Traffic Monitoring

The malware may sniff all TCP and UDP traffic on the machine. This is achieved through the use of RAW sockets. The following status strings can be sent to the C&C server:

```
Initializing Winsock 2.2...
Creating RAW socket...
Configuring socket for packet interception
Starting the sniffing process...
```

```
UDP Packet Information:
Source IP: %s DESTINATION IP: %s
SOURCE PORT: %d DESTINATION PORT: %d
PACKET DATA:
```

```
#####
#####
```

```
TCP Packet Information:
```

```
Source IP: %s DESTINATION IP: %s
SOURCE PORT: %d DESTINATION PORT: %d
```

```
#####
#####
```

If the current user privileges are insufficient for such action, the following error is presented:

```
the processs is not admin try after restart to while mpkProcess To Admin...
```

File Exfiltration

The malware may exfiltrate any file to the remote C&C server. The malware also contains the ability to enumerate all files on the system or find a specific file with the required filename specified by the operator.

Upon file exfiltration, the file is checked for size. This is performed in order to send the file in 4Kb 'chunks', where each chunk frame is sized 0x1014h bytes.

Before uploading any file to the C&C server, the malware will report its size:

```
length: %d
```

After sending each chunk, the malware will report the current transfer status:

```
%d Bytes / %d Bytes
```

When the transfer will be finished, it will report completion using the following string:

```
Completed: %d Bytes Downloaded.
```

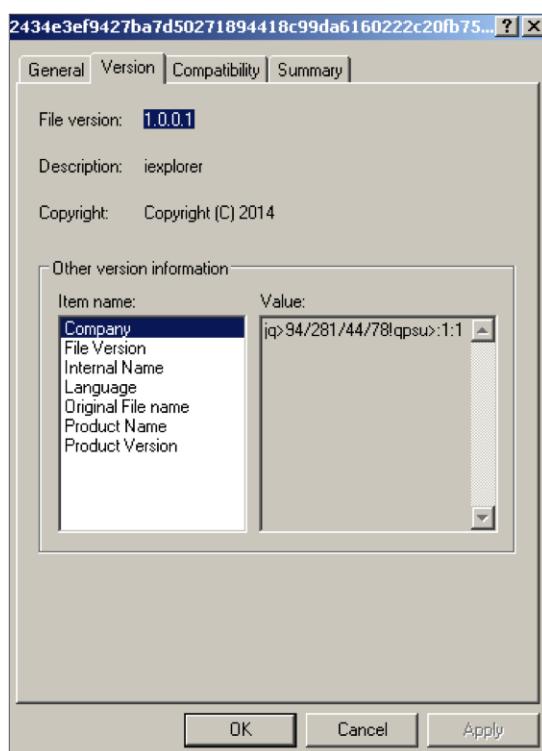
If there was any problem, this string will be reported:

```
Failed to open %s, %s not found.
```

Communication protocol

The malware uses raw sockets over IP protocol (IPPROTO_IP flag), effectively implementing its own protocol for data transfer.

The executable's own "File Version Info" is parsed to retrieve the server IP, trivially encoded into the "Company" value:



This data contains hardcoded IP address and port of C&C server. In this sample it is:

83.170.33.67:9090

A connection will be established to that IP, while sending periodic 'keep-alive' messages, containing these 6 bytes:

123456

File-exfiltration packets are 0x1014h bytes long. The first 2 bytes indicate the type of file to be exfiltrated:

- 0811h-logs (initial packet)
- 0810h-logs (subsequent packets)
- 080Fh-logs (final packet)
- 0BCDh-webcam images (initial packet)
- 0BCFh-webcam images (subsequent packets)
- 0BCEh-webcam images (final packet)
- 0803h-screenshots (initial packet)
- 0805h-screenshots (subsequent packets)
- 0804h-screenshots (final packet)
- 13C2h-error with file

The filename is located at offset 0x08h of the first packet. Subsequent packets include file contents only.



The Check Point Incident Response Team is available
to investigate and resolve complex security events
that span from malware events, intrusions or denial of service attacks.

The team is available 24x7x365 by contacting
emergency-response@checkpoint.com
or calling 866-923-0907