Scarlet Mimic: Years-Long Espionage Campaign Targets Minority Activists

89,832 people reacted 12 27 min. read

SHARE

By Robert Falcone and Jen Miller-Osborn

January 24, 2016 at 5:00 PM
Category: Reports, Threat Prevention, Unit 42
Tags: Android, Apple, AutoFocus, BrutishCommand, CallMe, cyber espionage, Cyber Threat Alliance, cybersecurity, Espionage, FakeM, Mac OS X, microsoft, MobileOrder, Psylo, Scarlet Mimic, SkiBoot Loader, SubtractThis, Trojans, WildFire

This post is also available in: 日本語 (Japanese)

Executive Summary

Over the past seven months, Unit 42 has been investigating a series of attacks we attribute to a group we have code named “Scarlet Mimic.” The attacks began over four years ago and their targeting pattern suggests that this adversary’s primary mission is to gather information about minority rights activists. We do not have evidence directly linking these attacks to a government source, but the information derived from these activities supports an assessment that a group or groups with motivations similar to the stated position of the Chinese government in relation to these targets is involved.

The goal of this report is to expose the tools, tactics and infrastructure deployed by Scarlet Mimic in order to increase awareness of this threat and decrease its operational success through deployment of prevention and detection countermeasures. From our vantage point, we are not able to identify which attacks have been successful against which organizations. But the fact that the tools Scarlet Mimic deploys have been under development for years suggests an active adversary that has been successful in some percentage of its operations. Based on our analysis, we are also seeing Scarlet Mimic start to expand its espionage efforts from PCs to mobile devices, marking an evolution in its tactics.

Individuals and groups of all different types may become the target of cyber espionage campaigns. The most well known victims of cyber espionage are typically government organizations or high-tech companies, but it’s important to recognize that espionage-focused adversaries are tasked to collect information from many sources.

The attacks we attribute to Scarlet Mimic have primarily targeted Uyghur and Tibetan activists as well as those who are interested in their causes. Both the Tibetan community and the Uyghurs, a Turkic Muslim minority residing primarily in northwest China, have been targets of multiple sophisticated attacks in the past decade. Both also have history of strained relationships with the government of the People’s Republic of China (PRC), though we do not have evidence that links Scarlet Mimic attacks to the PRC.

Scarlet Mimic attacks have also been identified against government organizations in Russia and India, who are

responsible for tracking activist and terrorist activities. While we do not know the precise target of each of the Scarlet Mimic attacks, many of them align to the patterns described above.

The Scarlet Mimic attacks primarily center around the use of a Windows backdoor named “FakeM.” It was first described by Trend Micro in 2013 and was named FakeM because its primary command and control traffic mimicked Windows Messenger and Yahoo! Messenger network traffic to evade detection. We have identified two subsequent variants of the FakeM family, which has undergone significant changes since it was exposed in 2013. We have also identified nine distinct “loader” malware families, which Scarlet Mimic appears to use to avoid detection when infecting a system.

In addition to the FakeM variants, Scarlet Mimic has deployed Trojans that target the Mac OS X and Android operating systems. We have linked these attacks to Scarlet Mimic through analysis of their command and control (C2) infrastructure.

To infect individuals with access to the data the actors desire, Scarlet Mimic deploys both spear-phishing and watering hole (strategic web compromise) attacks. Using these tactics they can directly target previously identified individuals (spear phishing) as well as unidentified individuals who are interested in a specific subject (watering hole). In their spear phishing attacks, Scarlet Mimic has exploited five separate vulnerabilities. However, in many cases they chose to forgo exploiting a software vulnerability and used self-extracting (SFX) RAR archives that use the Right-to-Left Override character to mask the true file extension, tricking victims into opening executable files.

As with many other attackers who use spear-phishing to infect victims, Scarlet Mimic makes heavy use of “decoy” files. These are legitimate documents that contain content relevant to the subject of the spear phishing e-mail. After the system is infected, the malware displays the decoy document to trick the user into believing nothing harmful has occurred. These decoy documents allow us to identify the theme of the spear phishing e-mail and in some cases the target of the attack.

The most recent Scarlet Mimic attacks we have identified were conducted in 2015 and suggest the group has a significant interest in both Muslim activists and those interested in critiques of the Russian government and Russian President Vladimir Putin. Based on their previous targets we suspect these individuals may be targeted based on the information they posses on activist groups.

The primary source of data used in this analysis is Palo Alto Networks WildFire, which analyzes malware used in attacks across the world. The system also analyzes malware samples collected through a sharing partnership with other security vendors, including our partners in the Cyber Threat Alliance. To connect attacks to each other based on malware behavior and command and control infrastructure, we relied on AutoFocus threat intelligence. AutoFocus users can view all of the files related to Scarlet Mimic and the malware associated with the group using the following links:

- ScarletMimic
- FakeM
- Psylo
- MobileOrder

Introduction

The better we can understand the threats to our networks and systems, the more effective we will be at preventing those threats. The goal of this report is to help network defenders better understand attacks from a group we have named Scarlet Mimic. This group has been conducting attacks for at least four years using a backdoor Trojan that has been under active development. The group primarily deploys spear-phishing e-mails to infect its targets, but was also responsible for a watering hole (strategic web compromise) attack in 2013.

Attacks from this group have been reported publicly in the past, but mostly as disparate, unconnected incidents. Based on analysis of the data and malware samples we have collected, Unit 42 believes the attacks described herein are the work of a group or set of cooperating groups who have a single mission, collecting information on minority groups who reside in and around northwestern China. In the past, Scarlet Mimic has primarily targeted individuals who belong to these minority groups as well as their supporters, but we've recently found evidence to indicate the group also targets individuals working inside government anti-terrorist organizations. We suspect these targets are selected based on their access to information about the targeted minority groups.

In the following sections we will describe selected attacks we have identified and who their likely targets are. We will also provide detailed analysis of the latest variants of the malware they deploy (known as FakeM) as well as other associated tools that allow Scarlet Mimic to target Android and OS X devices.

Attacks launched by this group were publicly exposed on 2013 in a Trend Micro report about the FakeM Trojan. Since that reports release, Scarlet Mimic has deployed two additional versions of the malware. They have also deployed nine separate "loader" Trojans they use to infect systems with their backdoor.

Attack Details

The majority of attacks we associate with Scarlet Mimic follow the pattern shown in Figure 1.

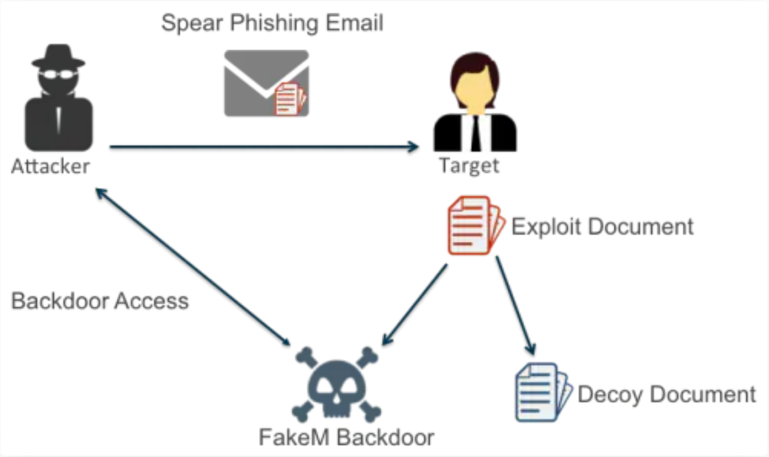


Figure 1: "Spear Phishing with Decoy" Attack Pattern Deployed by Scarlet Mimic

The attacker sends a spear-phishing e-mail with a subject and body content that appeal to the targeted user. This e-mail carries an attachment, which is typically a document that exploits a Microsoft Office vulnerability. The attachment uses a file name that is related to the e-mail content to trick the user into opening it. If the user opens the file and the exploitation is successful, a backdoor Trojan is installed on the system that gives the attacker access and a decoy document is displayed to the victim. Decoy documents are typically non-malicious versions of the content the user expected to see when opening the attachment.

Many of the targets and spoofed or compromised sending e-mail addresses have contact information on the Internet. The apparent sender email usually appears to be someone associated with the accompanying text, when appropriate, while the target emails are usually also available online tied to target organizations. A small subset of the decoys could not be found online and may be from previous compromises by Scarlet Mimic.

Many attackers deploy this particular pattern, as it is often successful at infecting a user without alerting the user of the infection. This is the exact same pattern, for example, deployed by the attackers in Operation Lotus Blossom.

We have identified spear phishing documents from Scarlet Mimic exploiting the following vulnerabilities.

- CVE-2012-0158
- CVE-2010-3333
- CVE-2010-2883
- CVE-2010-2572
- CVE-2009-3129

We also know Scarlet Mimic uses a number of toolkits to create documents that contain exploit code to install the FakeM payload on a compromised system. Unit 42 tracks the toolkits delivering FakeM under the names MNKit, WingD and Tran Duy Linh. These kits appear to be used by many attack groups, and they alone are not a good indication of Scarlet Mimic activity.

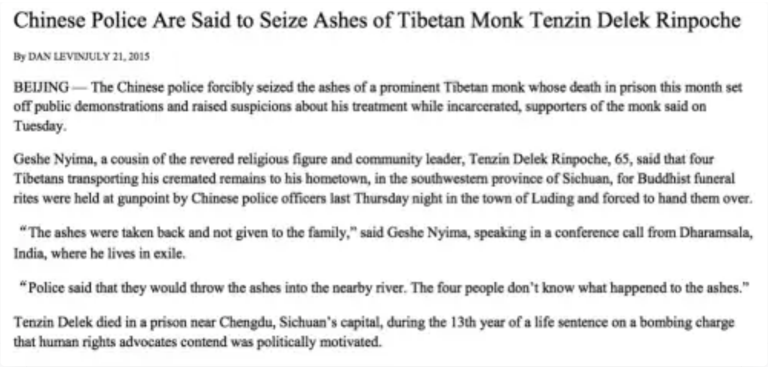
Additionally, in many cases these threat actors did not use an exploit document at all, rather they sent self-extracting (SFX) RAR archives that use the Right-to-Left Override character to mask the true file extension. For example, the following two filenames of SFX archives used to deliver FakeM contain the RLO character (bolded):

*Update about the status of Tenzin Delek Rinpoche'ashes**%E2%80%AE**cod.scr*
*tepsiliy mezmun.**\xe2\x80\xae**txt.scr*

Even when no software vulnerability is exploited, the attacks still typically include a decoy document. The content of most of the decoy documents appear to be available on the open Internet, and the attackers typically made small modifications to them.

Many of the targets and spoofed or compromised sending accounts have contact information on the Internet. The apparent sender email usually appears to be someone associated with the accompanying text, when appropriate, while the target emails are usually also available online tied to target organizations. A small subset of the decoys could not be found online and may be from previous compromises by Scarlet Mimic. The overarching decoy themes were Uyghur-related, anti-Putin, or Al-Qaeda-related. The decoys are often copied from think tanks or reputable news sources the targets would likely frequent.

In one instance, the threat actors used content from a *New York Times* article (Figure 2) on the same day it was published.



Chinese Police Are Said to Seize Ashes of Tibetan Monk Tenzin Delek Rinpoche

By DAN LEVINJULY 21, 2015

BEIJING — The Chinese police forcibly seized the ashes of a prominent Tibetan monk whose death in prison this month set off public demonstrations and raised suspicions about his treatment while incarcerated, supporters of the monk said on Tuesday.

Geshe Nyima, a cousin of the revered religious figure and community leader, Tenzin Delek Rinpoche, 65, said that four Tibetans transporting his cremated remains to his hometown, in the southwestern province of Sichuan, for Buddhist funeral rites were held at gunpoint by Chinese police officers last Thursday night in the town of Luding and forced to hand them over.

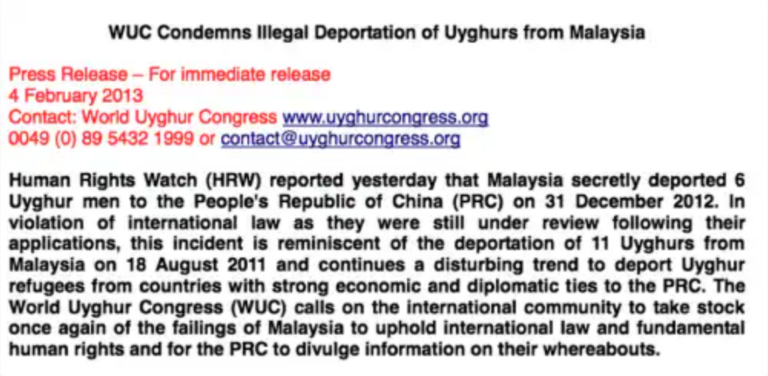
"The ashes were taken back and not given to the family," said Geshe Nyima, speaking in a conference call from Dharamsala, India, where he lives in exile.

"Police said that they would throw the ashes into the nearby river. The four people don't know what happened to the ashes."

Tenzin Delek died in a prison near Chengdu, Sichuan's capital, during the 13th year of a life sentence on a bombing charge that human rights advocates contend was politically motivated.

Figure 2: Decoy Text Extracted from the New York Times article

Figure 3 shows one of the more common themes used to target Uyghurs and those interested in their cause. Multiple attacks used press releases or other content related to the World Uyghur Congress.



WUC Condemns Illegal Deportation of Uyghurs from Malaysia

Press Release – For immediate release
4 February 2013
Contact: World Uyghur Congress www.uyghurcongress.org
0049 (0) 89 5432 1999 or contact@uyghurcongress.org

Human Rights Watch (HRW) reported yesterday that Malaysia secretly deported 6 Uyghur men to the People's Republic of China (PRC) on 31 December 2012. In violation of international law as they were still under review following their applications, this incident is reminiscent of the deportation of 11 Uyghurs from Malaysia on 18 August 2011 and continues a disturbing trend to deport Uyghur refugees from countries with strong economic and diplomatic ties to the PRC. The World Uyghur Congress (WUC) calls on the international community to take stock once again of the failings of Malaysia to uphold international law and fundamental human rights and for the PRC to divulge information on their whereabouts.

Figure 3: Decoy using World Uyghur Congress Press Release

In July of 2015, we identified a full e-mail uploaded to an antivirus scanning service that carried a Scarlet Mimic exploit document. In this case (Figure 4) the recipient of the e-mail was an individual working for the Russian Federal Security Service (fsb.ru). The e-mail body requests help dealing with threatening phone calls from an international gang.

Figure 4: Phishing E-mail send to FSB E-mail Address

Another attack, sent to an unknown target carried a decoy image (Figure 5) that compared Russian President Vladimir Putin to Adolf Hitler.

Figure 5: Anti-Putin image used as a decoy document

In yet another case, the threat actors used a conference notification from one organization (Figure 6a) and modified it to appear as thought it was for an "Islamic Country Muslim Religion Conference" (Figure 6b). This document was particularly poorly altered, as the registration form still contained multiple hints to indicate the document was fraudulent (Figure 6c).

Figure 6a: Original document used as a Psylo decoy

Figure 6b: Modified header and contact email in the decoy

Figure 6c: Bottom of the decoy document with replaced email and non-altered date -- a quick search online shows this to be fraudulent

In total we have collected over 40 individual decoy documents used in these attacks, far more than we can detail here.

We are aware of one case where Scarlet Mimic broke from the spear-phishing pattern described above. In 2013, the group deployed a watering hole attack, also known as a strategic web compromise to infect victims with their backdoor. The watering hole is an attack vector that involves compromising a website that targeted victims are likely to visit in order to infect and gain access to their systems. According to a blog by Websense, threat actors compromised the Tibetan Alliance of Chicago's website to host malicious code that exploited a vulnerability in Internet Explorer (CVE-2012-4969.) Microsoft patched this vulnerability in September 2012, suggesting that this watering hole attack used an older vulnerability, which aligns with the threat groups continued use of older vulnerabilities in their spear-phishing efforts.

Malware Overview

First discussed in January 2013 in a Trend Micro whitepaper, FakeM is a Trojan that uses separate modules to perform its functionality. FakeM's functional code is shellcode-based and requires another Trojan to load it into memory and execute it. There are a variety of different Trojans used to load FakeM, some of which are more interesting than others. In this section, we will explore the loader Trojans followed by an analysis of the evolution of FakeM itself. We end this section with a discussion on tools related to FakeM and used by Scarlet Mimic.

Loader Trojans

FakeM is shellcode-based and therefore requires another Trojan to load FakeM into memory and execute its functional code. Threat actors have developed many different loading Trojans to load FakeM, some of which are fairly straightforward while others use very clever techniques to avoid detection. Unit 42 tracks the following list of loader Trojans that Scarlet Mimic has used to execute FakeM:

- CrypticConvo
- SkiBoot
- RaidBase
- FakeHighFive
- PiggyBack
- FullThrottle
- FakeFish

- BrutishCommand
- SubtractThis

It appears that the threat actors include the loader Trojans in some sort of builder application that allows actors to quickly create, configure and deploy payloads to execute FakeM. We believe this because many samples that execute FakeM have the same exact compilation time but different C2 servers, as seen in the example in Table 1. This suggests the actors compile a single sample and use a builder tool to configure individual samples on demand.

We used the loader Trojans to provide a general timeline for the development of FakeM samples, as FakeM is shellcode-based and does not contain any usable timestamps. The timestamps in the loader Trojans does not necessarily correspond to the usage of FakeM, but plotting the compile times of the loaders on a timeline shows an interesting trend. The scatter plot timeline in Figure 7 shows the known compilation times of the loader Trojans and the FakeM variant that it executed.

SHA256	Compiled	Loader Trojan	C2 Domains
5182dc8667432d76a276dc4f864cdfcef3e481783ebaf46d3b1397080b798f4a	2013-09-13 08:02:58	CrypticConvo	opero.spdns[.]org, firefox.spdns[.]de
5dade00db195087aa336ce190b5fd1c22992c49556c623b42a9f742d73241a7f	2013-09-13 08:02:58	CrypticConvo	intersecurity.firewall-gateway[.]com

Table 1: Two samples sharing a compile time yet contain different C2 domains in their configurations

Figure 7: Timeline of compilation of loader Trojans associated with FakeM

Based on the timeline, it appears that the actors were actively developing several of the loaders at the same time from 2009 until the early months of 2014. After the first quarter of 2014, it appears that the actors abandoned development of the older loaders in favor of FakeFish, BrutishCommand and SubtractThis. This does not mean that actors will not continue to use the older loaders, but it does suggest that the actors will continue including the newer or freshly developed loaders in updated builder applications.

The timeline also presents the possibility that the FakeM developers reacted to the release of Trend Micro's FakeM blog and whitepaper. Trend Micro published their analysis of the FakeM Trojan on January 17, 2013 (marked in Figure 7 by a red line) that discussed the original variant of FakeM. Shortly after, the original variant of FakeM drops off the timeline in favor of the SSL and Custom SSL variants. It is possible that the FakeM developers saw their tool was exposed and adapted it to avoid detection for continued use as a payload in attacks. We cannot be certain if the developers reacted specifically to Trend Micro's content, as it is possible that they were reacting to the increased antivirus detection rate of their tool that resulted from the exposure of the tool. Regardless of the specific stimulus, the reaction shows that the FakeM threat actors evolved to avoid detection/attribution and to continue their attack campaigns.

The timeline does have one noticeable outlier, specifically the FakeHighFive sample compiled in September 2009 that loaded a FakeM SSL sample. We believe this compile time is incorrect, as the C2 domain for this sample, specifically press.ufoneconference[.]com, was registered by the threat actor in February 2013. The registration of the C2 domain in February 2013 aligns with other compilation times of FakeM SSL, which leads us to the conclusion that the September 2009 compilation timestamp was modified and/or inaccurate.

Most of the related loader Trojans, such as CrypticConvo, PiggyBack, FullThrottle, FakeHighFive, FakeFish and RaidBase do little more than load encrypted FakeM shellcode (either from a PE resource or embedded data), decrypt it, and execute the resulting shellcode. Other related loading Trojans, such as SubtractThis, BrutishCommand and SkiBoot employ clever techniques worth discussing.

SubtractThis

The SubtractThis loader displays a technique that is quite clever. This loader received its name based on a technique it uses to delay before carrying out its main functionality, specifically by requiring the user to hit the minus ("-") key. SubtractThis carries out this technique through the following steps:

1. Calls LoadAcceleratorsA function to load the virtual key for the minus character "-". Example: LoadAcceleratorsA(hInstance, VK_SUBTRACT);
2. Calls SetTimer function to set up a callback function that will be called in the event that the LB_FINDSTRING Windows message. Example: SetTimer(0, LB_FINDSTRING_, 10000u, TimerFunc);
3. Creates a continuous loop that starts by calling GetMessageA to obtain Windows messages
4. Calls TranslateAcceleratorA to check Windows message received is VK_SUBTRACT "-".
5. Calls the callback function set up in the SetTimer function if the user enters the minus "-" key.

This technique requires user interaction, which makes analysis in sandboxes more difficult.

BrutishCommand

The BrutishCommand loader uses a very interesting method to decrypt the FakeM functional code. The main function in this loader checks the command line arguments passed to it, and if there are none present it will obtain a random number between 0-9 and create a new process using the same executable with this random number as a command line argument.

If the executable has a command line argument, the Trojan subjects the value to a hashing algorithm and compares the hash to 0x20E3EEBA. If the value matches the static hash, the executable will subject the command line argument to a second algorithm that will produce a value that the Trojan will use as the decryption key to decrypt the embedded FakeM shellcode. It essentially brute forces its own decryption key by rerunning itself over and over until it runs with the correct value is provided on the command line. Unit 42 had not seen this technique used by other malware families and it introduces a challenging hurdle when attempting to analyze or debug the loader Trojan.

SkiBoot Loader

SkiBoot reads the master boot record (MBR) of the system to determine the XOR key that it will use to decrypt the FakeM shellcode. It carries out this functionality by calling the ReadFile function to read 512-bytes from “\\.\PHYSICALDRIVE0” and specifically uses the last byte of the MBR as the encryption key. The last byte of the MBR is “\xAA”, or the second byte of “\x55\xAA”, which is the boot signature portion of the MBR.

Instead of using ReadFile, one variant of this loader reads the MBR using DeviceIOControl using the ID_CMD control code, and accesses a specific offset to obtain the value that it will rotate each byte in the ciphertext within the decryption algorithm. The significance of using DeviceIOControl is that the VMware hypervisor responds to this API call with a blank buffer instead of the MBR, whereas the Virtualbox hypervisor returns the MBR correctly. It appears that this loader is specifically using the DeviceIOControl API function as a VMware detection technique, suggesting that the developers are well versed in the nuances of the VMware hypervisor and virtual machine evasion.

Evolving FakeM: Variants

Since being originally exposed in 2013, authors of FakeM have continuously made changes to the FakeM codebase, resulting in multiple variants. Before elaborating on the different variants of FakeM, there are many similarities that remain throughout the various iterations. The architecture has not changed during the evolution of FakeM, as a modular framework exists in each variant, as seen in Figure 8. The FakeM main module is responsible for launching embedded modules, such as a keylogger or for gathering sensitive files. The main module is also responsible for communicating with its C2 servers and handling commands issued by the C2 server.

Figure 8: FakeM Architecture

All FakeM variants initiate communications with its C2 server and check the C2’s response for a command. Also, all FakeM variants share a common command handler with the same capabilities, as seen in Table 2. The limited command set suggests that FakeM’s functionality is obtained by additional assembly code provided by the C2 server with the 0x211 command. According to Trend Micro’s initial analysis on FakeM, threat actors delivered and ran additional code that provided further capabilities to the Trojan, such as the ability to run shell commands, steal passwords, capture the screen and upload files.

Command	Description
0x211	Run assembly code directly from the C2.
0x212	Idle. Attempts to receive another command immediately instead of sleeping for 30 seconds.
0x213	Sets a flag to end the session with the C2. This will force the Trojan to reestablish a session with the C2.
0x214	Exit process.

Table 2: Command handler within all variants of FakeM

Now that we have covered the commonalities between FakeM variants, the following sections will dive into the specific variants of FakeM. Unit 42 categorizes the different variations of FakeM based on the method used to communicate with the C2 server, which has changed dramatically over the years.

Original FakeM

The original variant of FakeM generates network beacons to its C2 server that begin with a 32-byte header that in most cases is meant to blend into network traffic generated by legitimate applications. Following this 32-byte header, the original variant of FakeM includes data encrypted using a custom encryption cipher that uses an XOR key of “YHCRA” and bit rotation between each XOR operation.

The original variant includes the FakeM discovered and published by Trend Micro in 2013, in which the authors of FakeM first attempted to evade detection of its C2 communications by pretending to be generated by legitimate messenger applications, such as MSN and Yahoo! messengers. Figures 9 and 10 show FakeM attempting to resemble MSN or Yahoo! Messenger traffic, as the first 32-bytes contain data that resemble legitimate traffic generated by these chat programs.

Figure 9: FakeM using fake MSN messenger traffic for C2 communication

Figure 10: FakeM using fake Yahoo! Messenger for C2 communication

In addition to emulating chat programs, FakeM has also included HTML code within the 32-byte header. As you can see in Figure 11, the overall structure of the beacon did not change, rather the only differences is the data in the header contains HTML tags. The HTML data in the header led Unit 42 to a whitepaper published by Malware.lu that suggested the MSN, Yahoo, and HTML versions of the original variant of FakeM all share a common server application that the threat actors use to build samples and control infected systems.

Figure 11: FakeM HTML tags in C2 header

In October 2013, FireEye published a blog about a sample of FakeM that did not use fake messenger or HTML data in the first 32 bytes of the C2 traffic, but instead used four repeating bytes to fill this portion of the packet, as seen in Figure 12. Unit 42 tracks this under the original variant, as it uses the same algorithm to encrypt the data and otherwise shares a common structure to the MSN, Yahoo, and HTML versions with the exception of the modification to the first 32 bytes.

Figure 12: FakeM C2 beacon with four repeating bytes

FakeM SSL

While performing infrastructure analysis on FakeM original variants, we came across shared infrastructure with domains that hosted C2 servers for malware samples that did not match the known FakeM communication protocols. Palo Alto Networks WildFire had analyzed many samples associated with these related C2 domains, all of which communicated with the C2 server using secure sockets layer (SSL). To determine the malware family that was generating this traffic, Unit 42 analyzed these samples and found that the functional code was the same as the original FakeM variant.

This discovery indicates the authors of FakeM introduced new code to the Trojan in order to use SSL to communicate with its C2 server. The drastic change in C2 channel warranted a new variant name, and we dubbed it "FakeM SSL". During the analysis of these samples we did not find any operational C2 servers to complete a handshake to establish a SSL session. During the handshake, the FakeM SSL samples will tell the server it supports 36 different cipher suites even though the samples appear to only support one. Unit 42 believes the cipher suite within the FakeM SSL variants uses Diffie-Hellman for key exchange and the RC4 cipher to encrypt the C2 communications.

FakeM Custom SSL

In July 2015, Scarlet Mimic delivered a spear-phishing email to a branch of the Russian government with intentions of installing a payload that was undetected by any antivirus vendors on VirusTotal. The high profile target and the lack of antivirus detection prompted Unit 42 to perform an in-depth analysis and found that it is yet another new variant of the FakeM Trojan. We also named this variant after it's communication protocol (FakeM Custom SSL.)

This new variant of FakeM shared the same functional code as its predecessors, but again the communications with the C2 dramatically differed from the other variants. Communications between this variant and the C2 server leverage what Unit 42 believes is modified SSL code, as the code is very similar to the FakeM SSL variant. The code appears to use Diffie-Hellman for key exchange and the RC4 algorithm like FakeM SSL; however, the initial packet sent to the C2 server did not contain a "client hello" message, which is required to initiate an SSL handshake. Instead, the initial packet sent data as seen in Figure 13.

Figure 13: Hexdump of FakeM custom SSL variant

This variant of FakeM skips the traditional SSL handshake, which involves an agreement on a cipher suite to use to encrypt communications. The FakeM code only supports one cipher suite, which makes the cipher suite agreement portion of the SSL handshake irrelevant. Instead, FakeM creates a session with its C2 server by exchanging keys. The lack of a valid handshake makes detection of this C2 stream difficult, as the packets sent between the Trojan and the C2 to establish this session contain random binary data. Network devices will also be unable to perform any SSL decryption due to the lack of detection and the inability to determine the cipher suite used to encrypt the data. Figure 14 below provides a visual depiction of the handshake procedure and the subsequent beacon and command messages.

Figure 14: Communications between system and C2 server, including the key exchange

The handshake starts with a key exchange, which the Trojan initiates by creating a 2048-byte buffer that it will store its key (128-bytes, followed by null values as seen in Figure 13) and sending its key to the C2 server. The C2 will respond with its own key (also 128-bytes, followed by null values) that the Trojan will store and use to encrypt future communications.

Once this key exchange is complete, the Trojan acknowledges the receipt of the server's key by sending another 2048-byte packet to the server. To build the acknowledgement packet, the Trojan creates a 2048-byte buffer filled with null values and copies the string "ws32.dll" to offset 8 and encrypts the buffer with the server's key.

After sending the acknowledgement packet, the Trojan will gather local system information and include it in a beacon to the C2 server. Like the packets sent in the key exchange, the beacon sent by the Trojan to the C2 server is 2048-bytes in length; however, the system information gathered by the Trojan is only 296 bytes followed by 1752 are null values to fill the 2048 byte buffer. The system information follows the 296-byte structure seen below:

```
struct beaconToC2 {
 DWORD static_value_130h;
 CHAR username[128];
 CHAR computername[128];
 DWORD static_value_70000h;
 DWORD os_codepage;
 DWORD campaign_code_1;
 DWORD campaign_code_2;
};
```

The Trojan will encrypt this data using the key provided by the C2 and send it to the server. The Trojan will then wait for the C2 to respond, which it will decrypt and parse for FakeM commands. Unit 42 has been unable extract any plug-ins from C2 network traffic; however, several FakeM custom SSL samples contain embedded plug-ins that run without interaction with the C2 server. Also, Unit 42 was able to extract several modules from the original FakeM server application, as seen in Table 3. All of these modules are shellcode-based plug-ins that would work with the custom SSL variant of FakeM with little to no modification.

MD5	Size (bytes)	Description
7a1410b2eceb99ec268b50e9371e74c1	3724	Process Plug-ins
092085e76512f071cab12f76ed09b348	2412	Shell Plug-ins
8f4cbb78356cb672bf2566e44315eb96	1768	File Plug-ins
16ab40f84fc47bab2c7874bb3164c5b4	2268	Screen Plug-ins
30337e99631a174d822dd3ea00a5f6cf	2204	Regedit Plug-ins
1f3fbb789bcbe9186a50c4f4db269736	1996	Service Plug-ins
4313d9d5fc6a090e2abc41633cb2c1fd	3196	HostInfo Plug-ins
fe75dff8b86dd8989d2ca00df19d51be	2220	KeyBoard Plug-ins
3e184a7af74905f3d3acbec913252f72	1884	Shell Plug-ins
b59e8751b9f61bd4f4b9b62de8242751	3896	OE Pwd Plug-ins
83ec457cba27e470404c942eb9242eeb	2156	U-Files Plug-ins

Table 3: Modules extracted from the original FakeM variant's server application

Related to FakeM Original: CallMe

CallMe is a Trojan designed to run on the Apple OSX operating system. This Trojan was delivered in targeted attacks on Uyghur activists in 2013 and used infrastructure associated with FakeM.

In February 2013, AlienVault performed analysis on the CallMe Trojan and found that it is based on a tool called Tiny SHell, an OSX shell tool whose source code is available on the Internet. The Trojan uses AES to encrypt the communication channel its C2 server, which will provide one of three commands to carry out activities on the compromised system, as seen in Table 4.

Command	Description
1	Get a file from the system and upload it to the C2 server.
2	Put a file on the system from the C2 server. File is saved to a specified filename in <HOME directory>/downloads/.
3	Create a reverse shell to interact with the compromised system.

Table 4: Commands Available in the CallMe OSX Malware

The infrastructure overlap between FakeM and CallMe involves the fully qualified domain name (FDQN) of "googmail.org", which was used by both FakeM and CallMe samples. This suggests that not only do these threat actors have the ability to compromise victims running the Microsoft Windows operating system, but they can also target individuals running Apple's OSX as well.

Related to FakeM Custom SSL: Psylo

During infrastructure analysis of FakeM Custom SSL variants, Unit 42 found infrastructure overlaps between FakeM and another new, previously unreported Trojan that we named “Psylo”. Psylo is a tool that allows threat actors to upload and download files to and from a compromised system, as well as execute commands and applications on the system. The name Psylo is an anagram from the mutex created when initially running this payload, which is 'hnxlopsyxt'.

Psylo is similar to FakeM in that they are both shellcode-based, and they have similar configurations and C2 communication channels. As you can see from the following two configuration structures, Psylo and FakeM have similar configurations with only the array length of the C2 locations being different.

struct psylo_c2_config {char[60] c2_host_1;char[60] c2_host_2;char[60] c2_host_3;DWORD c2_port_1;DWORD c2_port_2;DWORD c2_port_3;};	struct fakem_customssl_c2_config {char[64] c2_host_1;char[64] c2_host_2;char[64] c2_host_3;DWORD c2_port_1;DWORD c2_port_2;DWORD c2_port_3;};

Figure 15: Comparison between Psylo and FakeM custom SSL configurations

Both use SSL to communicate with their C2 servers, and it appears they share common code to carry out the Diffie Hellman key exchange. We compared the Diffie Hellman code from Psylo with FakeM custom SSL variant and found that they were very similar, but the FakeM samples had some of the functionality within sub-functions, which rendered binary diffing between the two Trojans impossible.

Another slight difference involves how Psylo and FakeM generate random numbers for SSL. FakeM uses QueryPerformanceCounter to create a random number, whereas Psylo uses CryptGenRandom, both of which generate random numbers 68 bytes long. Interesting enough is that CryptGenRandom calls RtlGenRandom, which uses QueryPerformanceCounter along with other system attributes to generate a random number.

When communicating with its C2 server, Psylo will use HTTPS with a unique user-agent of (notice the lack of a space between "5.0" and "(Windows"):

Mozilla/5.0(Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0

Unit 42 does not consider Psylo another variant of FakeM because Psylo has a command handler that differs dramatically from FakeM. Table 5 shows the Psylo command handler, which suggests it is less modular and supports more embedded functionality when compared to FakeM. It is possible that the threat actors created this Trojan as a standalone alternative to FakeM.

Command	Description
0	Idles for 10 seconds.
2	Enumerate all storage devices.
3	Find all files that starts with a particular string (%s*.*).
5	Creates a file to write to, deleting it if it already exists. Combined with 'E' command to download a file to the system.
E	Writes data from C2 to a file opened using the '5' command. Combined with '5' command to download a file to the system.
6	Reads a file, which effectively uploads the file to the C2.
7	Delete a specified file.
8	Execute a command using WinExec. Responds to C2 with 's' if successful or 'r' if unsuccessful.
9	Timestomps. Sets a specified file's timestamps to match that of a system file in the System32 directory.

Table 5: Command handler in Psylo that differs dramatically from FakeM

MobileOrder: Mobile Devices the Next Frontier

Another discovery we made while researching this blog is an overlap between Psylo infrastructure and a Trojan focused on compromising Android mobile devices. Unit 42 tracks this mobile Trojan as MobileOrder, as the authors specifically refer to commands within the app as orders. The connection between FakeM, Psylo, and MobileOrder suggest that Scarlet Mimic is now expanding their espionage efforts from PCs to mobile devices, which marks a major shift in tactics.

MobileOrder starts by registering itself as device administrator so that a normal user cannot uninstall it by simply clicking “uninstall” in settings. It will copy an embedded PDF file from “res/raw/rd.pdf” to SD card "/android/9074ca3f18e201c204ec1d852264bb5432644ba46f54f361a146957.pdf" and launches the mobile device’s default PDF viewer to display this PDF file, which acts as a decoy document. After displaying the decoy document, the malicious code runs in background. The malicious code consists of the following parts:

1. An Android geographical location SDK provided by AMAP.
2. Actor developed code that carries out Trojan's functionality.

The malware uses the AMAP SDK to get accurate location of infected devices by GPS, mobile network (such as base stations), WiFi and other information. MobileOrder acts on instructions provided by its C2 server, which it communicates with over TCP port 3728. All C2 communications are encrypted with the AES algorithm using a key generated by computing five MD5 hashes starting with the key "1qazxcvbnm", and adding a salt value of ".)1/" in each iteration.

The C2 server will respond to requests from MobileOrder with commands that the Trojan refers to as "orders". MobileOrder contains a command handler with functionality that provides a fairly robust set of commands, as seen in Table 6. The first byte of data provided by the C2 server is order number, which is followed by the encrypted data that needed to carry out the specific order.

Order #	Order Name	Behaviors
18	Order_Folder_List	Upload names and attributes of files under specified path
20	Order_Process_List	Upload all running processes information
24	Order_HostInfo	Upload device information including IMEI, IMSI, SIM card serial number, phone number, Android version, device manufacturer, device model, SD card size, network type, device locking status, country, carrier, time zone, language, install app list, browser bookmarks, etc.
26	Order_FileDelete	Delete specified file
27	Order_DownLoad	Download specified file to SD card's Android/data/tmp/ directory.
28	Order_UpFile	Upload specified file to C2 server
51	Order_Sms	Upload all received and sent SMS addresses, content, date, time to C2 server
52	Order_Contact	Upload all contacts' information to C2 server
53	Order_Call	Upload all phone calling history information
54	Order_Camera_front_photo	Take a picture by device's front camera
56	Order_SetSleepTime	Set sleep time interval
57	Order_SetOnline	Stop sleep
58	Order_SetMediaRecorder	Start audio recorder in specified time
59	Order_GetLoc	Upload information about network operator, MCC, MNC, network type, GSM cell location, CID, LAS, BSSS, etc. This information can be used to locate the device.
60	Order_GetGps	Upload GPS location by AMAP SDK.
61	Order_SetTelRecorderOn	Activate phone calling recording

62	Order_SetTelRecorderOff	Deactivate phone calling recording
81	Order_Install	Install specified APK file. May install silently or install to system app according to C2 command data
82	Order_Uninstall	Uninstall specified app
84	Order_StartApp	Launch specified app
85	Order_SendBroadcast	Send specified Android broadcast to launch other apps
86	Order_Shell	Execute specified shell commands
87	Order_OpenTrack	Start geolocation tracking in AMAP SDK
88	Order_CloseTrack	Stop geolocation tracking in AMAP SDK
90	Order_CheckScreen	Check whether phone screen is on (or said whether the phone is used by its owner)

Table 6: MobileOrder command handler

Infrastructure Overlap and Related Tools

There is some infrastructure overlap in the C2 servers used by almost all of the FakeM variants, as well other Trojans such as MobileOrder, Psylo, and CallMe. There are also infrastructure ties between some FakeM variants and older activity using Trojans such as Elirks, Poison Ivy, and BiFrost, which were used in attacks as old as 2009. The domain names used to host C2 servers are a mix of actor-registered and Dynamic DNS (DDNS,) though most are DDNS. The DDNS domains in turn are linked to a small grouping of ASNs, with one ASN often largely tied to one FakeM variant. Most of the FakeM MSN C2s resolve to IP addresses associated with ASN 22781 (RBLHST - Reliablehosting.com). However, we found one MSN sample that shared infrastructure with some FakeM Custom SSL variants.

There is a similar overlap between FakeM MSN, FakeM HTML, and FakeM SSL. The registrant email xsldmt@xj163[.]cn was used to register several domains used as C2s: yourturbe[.]org, websurprisemail[.]com and googmail[.]org. One of these domains was also used in the 2013 CallMe activity at the same time it was being used for FakeM MSN samples. The targeting and decoy style also matches with the FakeM activity.

There is PE resource overlap between some FakeM MSN samples and some samples of the BiFrost and Poison Ivy Trojan. This may indicate that the same developer who created the particular BiFrost and Poison Ivy samples was also involved in developing FakeM MSN. Unit 42 found an overlap between the RT_VERSION resources, which contains the version information of a Portable Executable (PE) file, shared amongst the three different Trojans. The shared RT_VERSION resource (MD5: 55b7a118203a831cc69b37b785015c54) contained the following information:

```
Comments: Release
CompanyName: Develop Team
FileDescription: Utility Application
FileVersion: 4.0
InternalName: Utility
LegalCopyright: Copyright (C) 2008
LegalTrademarks: DT.Inc
```

```
OriginalFilename: Utility.EXE
PrivateBuild: 4.0b
ProductName: Utility Application
ProductVersion: 4.0
```

The overlap between Elirks, FakeM SSL, Psylo, and MobileOrder exists entirely in their command and control infrastructure, through domain names and/or IP resolution. Samples of these three used some of the same C2 domains, notably lenovositegroup[.]com, ufoneconference[.]com, and websurprisemail[.]com, while some resolution overlap includes 118.193.212[.]12, 210.206.219[.]241, and 59.188.239[.]117. Similarly, some FakeM Yahoo C2 domains and FakeM Custom SSL C2 domains also have overlapping IP resolutions, notably 95.154.204[.]198.

Scarlet Mimic also uses the infamous HTRAN tool on at least some of their C2 servers. HTRAN is a proxying tool that allows actors to conceal the true location of their C2 server. Actors will run HTRAN on a server and configure their malware to interact with that server; however, the actor will configure HTRAN to forward traffic to another server where the actual C2 server exists. For example, the FakeM C2 domain of “muslim.islamhood[.]net”[1] resolved to the IP address 59.188.239.117 during analysis, but the server responded with the following error message:

[SERVER]connection to 68.71.35.135:8081 error

This error message suggests that the HTRAN application running on 59.188.239.117 was unable to connect to the real C2 server hosted at 68.71.35.135.

Prior Publications

Throughout this report, we have referenced multiple previous blogs and white papers, from Unit 42 and others, that have documented elements of this threat in the past. In addition to those documents readers may also find the following publications interesting.

In 2014, Citizen Lab released a paper on threats against civil society that referenced some of these attacks as the “Domain Name Family” or DNF.

Kaspersky Lab has produced excellent research on attacks against Uyghur and Tibetan activists. In 2013, they identified an Android Trojan that was also targeting these groups. Our analysis indicates this malware is different from the MobileOrder Trojan described above, but they serve very similar purposes.

On January 12, 2016, Cylance published a blog linking an exploit document to the group Mandiant refers to as APT2 and CrowdStrike as “Putter Panda.” While there does appear to be a small overlap between IP addresses used in attacks from this group and those of Scarlet Mimic, our team has not concluded that these groups are one in the same.

Conclusion

The information discovered by Unit 42 and shared here indicates Scarlet Mimic is likely a well-funded and skillfully resourced cyber adversary. Scarlet Mimic has carried out attacks using both spear-phishing and watering holes since at least 2009 with increasingly advanced malware, and has deployed malware to attack multiple operating systems and platforms. Despite the apparent technical acumen, their decoy documents are typically not well crafted regardless of the use of the target's language, though they do use timely subject lures.

The primary source of data used in this analysis is Palo Alto Networks WildFire, which analyzes malware used in attacks from around the globe. The system is also fed with malware samples collected through sharing partnership with other security vendors, including our partners in the Cyber Threat Alliance. To connect attacks to each other based on malware behavior and command and control infrastructure, we relied on Palo Alto Networks AutoFocus threat intelligence. AutoFocus users can view all of the files related to Scarlet Mimic and the malware associated with the group using the following links:

- ScarletMimic
- FakeM
- Psylo
- MobileOrder

Palo Alto Networks customers are protected from Scarlet Mimic attacks through many components of our platform.

- Threat Prevention signatures for the software vulnerabilities listed in this report are available to detect the exploit files during delivery.
- Traps, our advanced endpoint solution, can prevent the software vulnerabilities listed in this report from being exploited on a Windows host.
- WildFire classified all of the Android and Windows malware described in this report as malicious.
- We have released anti-malware signatures for the files listed in this report.
- The domain names used for command and control have been classified as malicious in PANDB.

Scarlet Mimic Indicator Data

FakeM Custom SSL Samples

```
12dedcdda853da9846014186e6b4a5d6a82ba0cf61d7fa4cbe444a010f682b5d
33e50c44804d4838dba6627b08210029ff9106fa7fd16cd7255271e153f58b05
3d9bd26f5bd5401efa17690357f40054a3d7b438ce8c91367dbf469f0d9bd520
5182dc8667432d76a276dc4f864cdfcef3e481783ebaf46d3b1397080b798f4a
523ad50b498bfb5ab688d9b1958c8058f905b634befc65e96f9f947e40893e5b
5dade00db195087aa336ce190b5fd1c22992c49556c623b42a9f742d73241a7f
7156f6416e7116e52f9c67f4e716b1dbea17387e61009c7f2825debbbb4dcb73
79aca57905cca1e56b0cedf48a4d81812639c333ee6532d90a074d64b3852d6f
879edf0417c4a9759040b51bf83b2fc918a6644a7c29a52252003a63036aea5c
9b77bbb620f50632fae17c40c7469fc93ffdbc4136a6d893a9a10a44bc435da5
a1b7fe2acdb7a5b0c52b7c1960cfad531a7ca85b602fc90044c57a2b2531699f
a268cc4931781d1d8094a4f8f596c2de3d662f2581c735b0810ff0ecefe3f859
a4abbcfdbf4a6c52349a843eac0396e6d8abb05f1324223980d824629a42ef7a
a569f3b02a4be99e0b4a9f1cff43115da803f0660dd4df114b624316f3f63dc6
b4c1e9c99f861a4dd7654dcc3548ab5ddc15ee5feb9690b9f716c4849714b20d
bbdedcfe789641e7f244700e8c028ef51094b66508f503876eb0d6aa16df6aa8
c7b9e6b5ab07e6da404af9894c8422d9a0c9586334ddc0a3c1ea6bf23ef97fb2
```

caeace73a17e220634525d2a4117525fd60cb86a06873c86571e89d156f8d72d
caf76e19a2681dd000c96d8389afc749e774c083aef09f023d4f42fbc49d4d3d
e96097826179a66cc3061be0f99f7b55cc9692a6378b5c4364699327823098ab
f511b13341c9fb4ec9ecfcfe5a5813b964c362d7c709c402ead4e010d857bf6c
fa08a498da0b31e77669d51a28dff166d84983fa6af693063c08f312fdce93e3
df9872d1dc1dbb101bf83c7e7d689d2d6df09966481a365f92cd451ef55f047d

FakeM SSL Samples

0aab09bf0db30a4be28d19475082fd5e7f75879bf9029fdd8dfc3a1e1f072b0c
2e1472a65a8df43c8bc9b0aff954fbc1a093c4214f6a718a08e1321db83ca683
3209ab95ca7ee7d8c0140f95bdb61a37d69810a7a23d90d63ecc69cc8c51db90
41948c73b776b673f954f497e09cc469d55f27e7b6e19acb41b77f7e64c50a33
4a4dfffae6fc8be77ac9b2c67da547f0d57ffae59e0687a356f5105fdddc88a3
5154511a439bb367b7dd56232eb15281cb6dc4d64ea3a06e7fbbe6b176e385d4
5fae5750797ebe7e7a6a6919a7d66deffb141ec28737bd72a1f7da8edd330b60
aa8a302a53bd39b2d2a6e3d8497575e2a5f9757b248e34c8e0821ce9eee5cc32
b3c9bb22fa1bc358dc23a1a4bdaf85ad1add4d812b107b7ab887affbf689933a
cd506679fd32dab16dee6fbf1cfdfe0836e092a4f5669418a199d99c9cd33abd
d1dd4469c7b5c462e5ff2dcef5d22775250e9ebf395f65da624f18ea7144e173
d698008e417da867d02e2f5cdcc80ff92af753dd585fada42fc611c2d7332c3a

FakeM Original Samples

53af257a42a8f182e97dcbb8d22227c27d654bea756d7f34a80cc7982b70aa60
9adda3d95535c6cf83a1ba08fe83f718f5c722e06d0caff8eab4a564185971c5
53cecc0d0f6924eacd23c49d0d95a6381834360fbbe2356778feb8dd396d723e
631fc66e57acd52284aba2608e6f31ba19e2807367e33d8704f572f6af6bd9c3
7bfbf49aa71b8235a16792ef721b7e4195df11cb75371f651595b37690d108c8
7c9421a4605decfa1b3e22addbca98d86ea757dcd8ff8e075d13228c99618637
202975d10ba417cf441e8f9986d2496807fe39e057d3226ec3b2713f0c218cd8
22e7517d8996e92998eb996416f9d8ef06b3b1c220c1a5d29ccd5aaef7b10c72
435df30d139ccbe5ce4e5ca6fe072e42e96d5ea1efd5317deebce462ecccc7ab
47d9ba5f7bf70c5d2b7a832e070957cc7ebdcfd0a6ee75851df16dc45971ce8a
4a3d0df9fa198a7ebe45db5239d22067e74924b1aace52029b3acc9b51af691e
4d539f638ed476ca08da838cdfbf710dae82b582256d60a009e9d304f6822e65
be0e8da7e261ec7d08eaa78e79ceb1be47c324b8e142097bf6569f9471c98a4e
c30d03750458bb5f2b03d6bd399ffca6d378a3adb5a74bee3b6ba4b982dbf273
cc7db456825e266849090b6fa95a94ad8c4c717712b610b0d39077af5222f4be
d6d2a77f8ed2fe9fed9ee6dcb4cc0b339ba47a575c717c35815243c752d8f60c
db8338e6b883fdceaa02c10ad683547a26ae32e0d4641cc24c7bd3b45154abb0
e8e5ecf525c5259651bfbdf1923215729ec67658225eca1b02519f5f6279eacb
ec4deb761b09ddc706804ef669836cf4b199f1d74b14ad623a6f6cc2f38190b8
669ce0975c133d54e414dbf1de546aed742e76fe3e60568e2bd4747b7e0f8b70
0d77f5f1d4c0f02fb88ac33fa365b17d28d1521cea59329ca4b3dd0b7031a60e
363d9557861fab2d83d04847b967996361e670e571b335c7a535bc6278cba149
7fb2c37431fd7b05414b134732ba0b29cd7dad17fc176627ee0815aac60c1ab9
77e4ef9e08f1095487b4fa27492b4c9b8e833f29598f99a0d10f7c85b4254761
a4ffca5f1c3d9c21629fa98a1e91121d954ab9347e86ac3c9613dae61bf30393
428121c421bf81a0d689014cf21ec7951b0c32add86198e06f7d636981f68755
a195f564aa2fb66db119e2fbec93e319a973e5cf50fbf9fc08bd81f9b7ee8af8
c1e8ff8ebe3754bc7d14509ef3678edf7551d876d3fa847d07d469c09bceae91

53238f67ac7e4cc27264efbacc8712bd97a5775feaf633c63adaa0785d038e8a
508a7cab0f2a69ba66e92e86817a49ecd1b9c8ae11a995147944995fc868dfad
fb60d14de4dba022f11437845d465a661d0c78d3d097a38770816f06992bf0af
8da2f9afd914a4318a97f4d74809c0c383f8ebf0d3d6e3d3715efbd71a66a52f
6fe33c672fd30bba9bbc89dc7d88993d8783382c9f9c510677b1bb068a5f1e51
6a1c7999b4ba92899d3364fc729d0f052680be5a71dd0f13cbabdb19b82bf858
5db51f2f7c31de7d165ec4892ae7dcedaa036caedeef718b57953d7935582f04
27167a9d63f5ddc68a12decb1a1e0a2a29c72fe0681dca2c4f3d169f048a9d38
6f10c892133b5dac6c40cfe77ca32b42572bc56909481b236080dfc143ef9afd
de12cd8d11478d17342c60239837c1afcc9fee72df6ffdf9943802640d43f77a
0f2db64b8283b76d49c9bb272beafab8323f941b6dc3888b42ff02f08634d016

Psylo Samples

19bbee954ac1a21595e63cb86d1a596236aed353804aec5cb8adfa62e70280d3
a9f0bddc3d3516af8355e8ac17309528cd018347e5f56a347c14da0a83b0028a
00bb399c429e0f1f7de751103fe92b5f820d1686d01662a08583b7a94aaed94e

MobileOrder Sample

03004ccc23033a09532bea7dfa08c8dfa85814a15f5e3aedb924a028bcd6f908

CallMe Samples

071c34b9701dd84f9590ba899a8af3eeec228a928f2d98a80dbc780e396ee01a
d1f0658bbb15ab2bccc210d7e1f21b96e14ae22de8494ca95b12e182f3d0f693
9ff687a813a5cb5ff10374c86f852534c1aa3e5a221123214bf52b2ff455a5da
8c423506c0c7ebe1e61071374ecf0806463a02a2100b5daa1bd942129ff8a235
91e36e720477146f1a0c050d3bc74bc6683a03e7631317ded3c598a10465dcc8
c981db20d588ba2d0f437b4e5459e7c6763f52a97841450c94591ca28a9a2d69
95dba004f949e44cb447246f3d2420b01db4541d0e4fa7b00d798f38a3d251e4

FakeM Custom SSL C2 Servers

aaa123.spdns[.]de
admin.spdns[.]org
detail43.myfirewall[.]org
economy.spdns[.]de
firefox.spdns[.]de
firewallupdate.firewall-gateway[.]net
intersecurity.firewall-gateway[.]com
kaspersky.firewall-gateway[.]net
kasperskysecurity.firewall-gateway[.]com
kissecurity.firewall-gateway[.]net
mail.firewall-gateway[.]com
news.firewall-gateway[.]com
opero.spdns[.]org
sys.firewall-gateway[.]net

FakeM SSL C2 Servers

account.websurprisemail[.]com
addi.apple.cloudns[.]org
bailee.alanna.cloudns[.]biz
bee.aoto.cloudns[.]org
book.websurprisemail[.]com
desk.websurprisemail[.]com
dolat.diyarpakzimin[.]com
dolat.websurprisemail[.]com
dolet.websurprisemail[.]com
github.ignorelist[.]com
islam.youtubesitegroup[.]com
mareva.catherine.cloudns[.]us
muslim.islamhood[.]net
p.klark.cloudns[.]in
ppcc.vasilevich.cloudns[.]info
press.ufoneconference[.]com
vip.yahoo.cloudns[.]info

FakeM Original C2 Servers

207.204.225[.]117
accounts.yourturbe[.]org
addnow.zapto[.]org
bits.githubs[.]net
clean.popqueen.cloudns[.]org
economy.spdns[.]eu
eemete.freetcp[.]com
email.googmail[.]org
fish.seafood.cloudns[.]org
freeavg.sytes[.]net
freeonline.3d-game[.]com
ibmcorp.slyip[.]com
lemondtree.freetcp[.]com
liumingzhen.myftp[.]org
liumingzhen.zapto[.]org
n.popqueen.cloudns[.]org
news.googmail[.]org
oic-oci.3-a[.]net
polat.googmail[.]org
qq.ufoneconference[.]com
qq.yourturbe[.]org
sisiow.slyip[.]com
update.googmail[.]org
uprnd.flnet[.]org
video.googmail[.]org
webmail.yourturbe[.]org
worldwildlife.effers[.]com
www.angleegg.ddns[.]us
www.angleegg.xxxy[.]info
www.googmail[.]org
youturbe.co[.]cc

yycc.mrbonus[.]com
zjhao.dtdns[.]net

Psylo C2 Servers

apple.lenovositegroup[.]com
mm.lenovositegroup[.]com
ftp112.lenta.cloudns[.]pw
www.gorlan.cloudns[.]pro
otcgk.border.cloudns[.]pw

MobileOrder C2 Servers

ziba.lenovositegroup[.]com

CallMe C2 Servers

apple12.crabdance[.]com
update.googmail[.]org
apple12.crabdance[.]com
alma.apple.cloudns[.]org

Popular Resources

Resource Center

Blog

Communities

Tech Docs

Unit 42

Sitemap

Legal Notices

Privacy

Terms of Use

Documents

Account

Manage Subscriptions

Report a Vulnerability