Banking Botnets Persist Despite Takedowns

APRIL 2015

Dell SecureWorks
Counter Threat Unit™ Threat Intelligence

Contents

Overview

Since the Dell SecureWorks Counter Threat Unit™ (CTU) research team published information about the top banking botnets of 2013, threats to banks and other financial institutions have grown and matured, and cybercriminals have become far more creative and increasingly organized. Although banks and financial institutions constantly improve their security measures to protect their online customers, the introduction of new malware families and the continual improvements to active malware campaigns pose challenges to the organizations and their customers.

Between mid-2014 and early 2015, coordinated efforts involving law enforcement and private-sector industry disrupted three of the most active banking botnets. Global law enforcement partnered with companies across national boundaries to launch two separate operations targeting the Gameover Zeus and Shylock botnets. In Operation Tovar, security researchers exploited design flaws in the Gameover Zeus peer-to-peer (P2P) network, disrupting the criminal infrastructure by manipulating the peer list and redirecting traffic to nodes under the researchers' control. A few weeks after Operation Tovar, another global operation led to the seizure of command and control (C2) servers and botnet-related domains associated with the Shylock infrastructure. In early 2015, Europol collaborated with multiple law enforcement and industry partners to seize servers and other important infrastructure owned by group behind the Ramnit botnet.

Cybercriminals quickly adapt to countermeasures and takedowns by improving their software and establishing new sophisticated banking botnets. New threats arise with emerging technologies, and attacks on mobile banking platforms and advancements in bypassing standard authentication mechanisms evolved in 2014.

Key findings

CTU analysis of banking botnet activity in 2014 and early 2015 revealed key findings:

- In addition to traditional banking websites, targets included websites for corporate finance and payroll services, stock trading, social networking, email services, employment portals, entertainment, hosting providers, phone companies, and dating portals.
- Attackers used banking trojans to target more than 1,400 financial institutions across more than 80 countries.
- More than 90 percent of banking trojans targeted financial institutions located in the U.S., but institutions in the UK, Germany, Italy, Spain, and Australia were also affected.
- Attackers focused on targets in Asian countries, where financial institutions implement weaker account security.
- Dyre, Bugat v5 (also known as Dridex), and Vawtrak (a Gozi variant) emerged after the Gameover Zeus and Shylock takedowns.
- Botnets increasingly rely on hidden network services such as Tor or the Invisible Internet Project (I2P), which resist surveillance and takedowns.
- Activity from Zeus and its variants decreased in the second half of 2014, while Dyre, Gozi/Vawtrak, and Bugat v5 activity steadily increased.
- Dyre and Bugat v5 incorporated private spam mailers, deviating from the "spam as a service" model used by other botnets.
- There was increased use of Kegotip, Chanitor, Upatre, and Lerspeng as first-stage downloaders/droppers.

Banking botnet activity

Traditionally, botnet owners protected their source code, often selling it for a high price when the owners retired to continue its operation. However, source code for botnets such as Zeus and Carberp have been leaked and used to develop new botnet variants. Between January 2014 and March 2015, CTU researchers observed banking botnet activity originating from the 13 botnets listed in Figure 1.

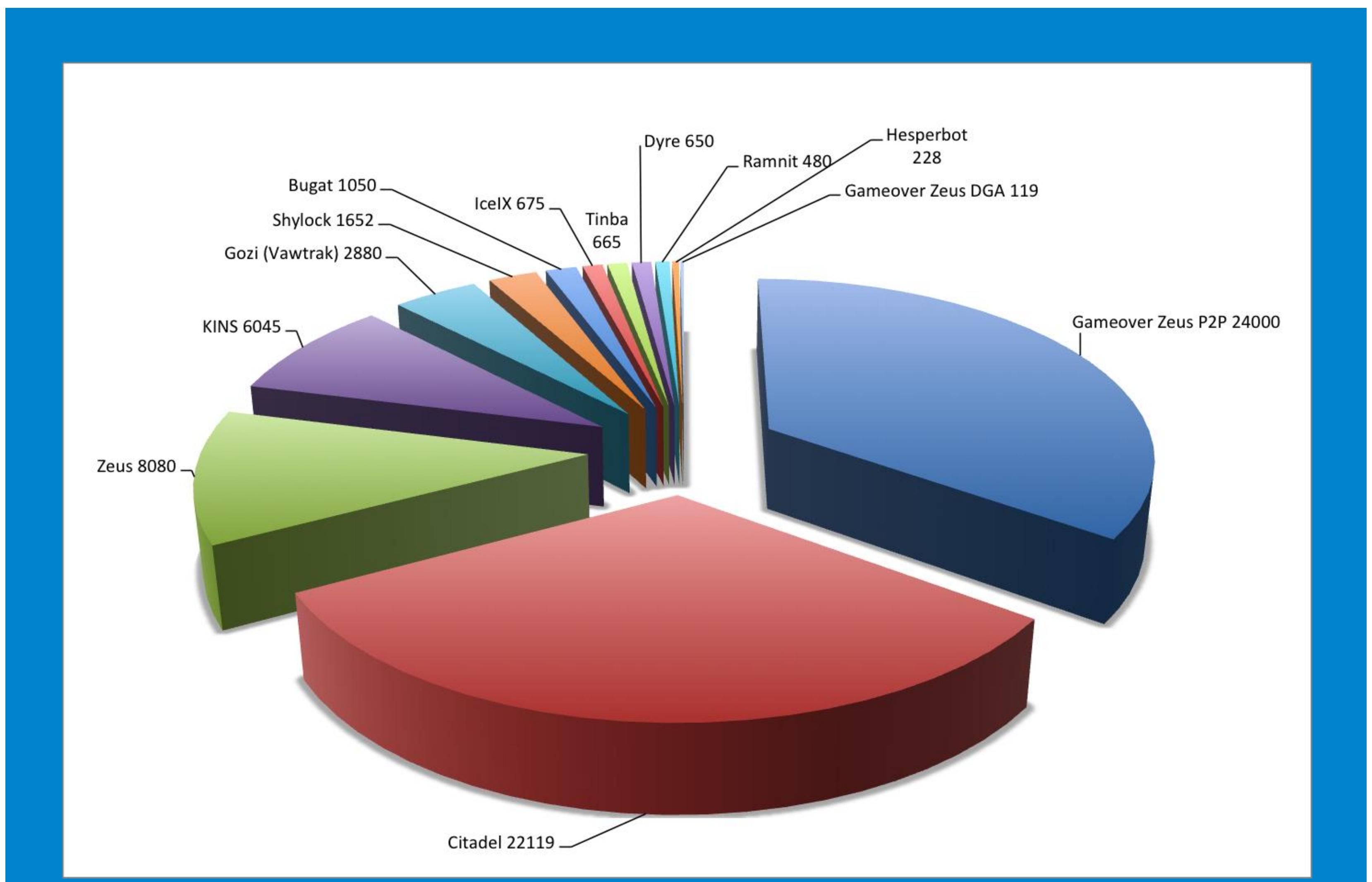


Figure 1

Prevalence of banking botnets between January 2014 and March 2015 based on the number of samples analyzed by CTU researchers. Gameover Zeus P2P and Shylock were active until June and July 2014, respectively. (Source: Dell SecureWorks)

Analysis of configuration files associated with these samples revealed that targets included the customers of more than 1,400 financial institutions. The banking botnets targeted commercial banks, credit unions, and other financial institutions in developed countries with sizeable populations and wealthy residents (see Figure 2).

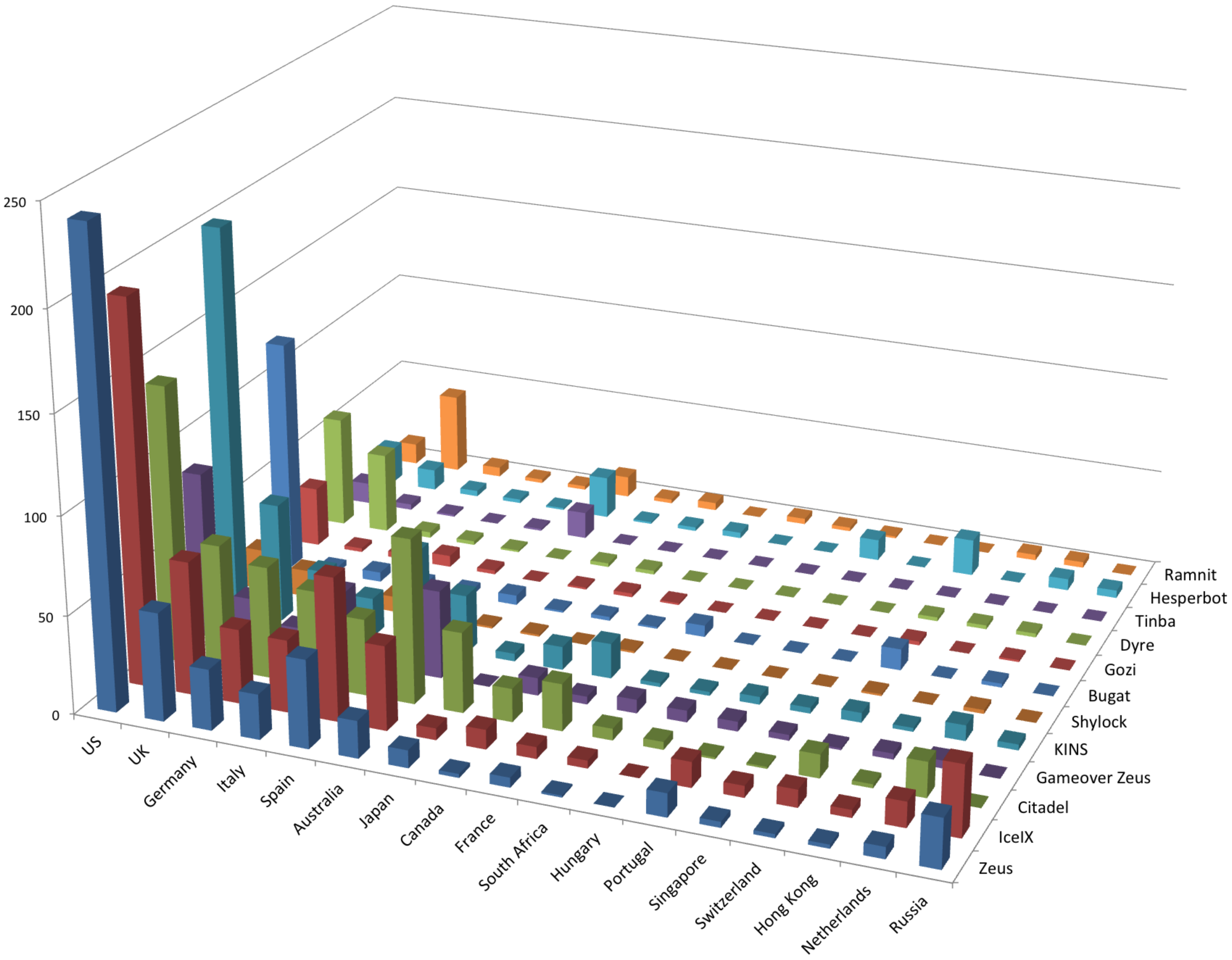


Figure 2

Countries targeted by banking trojans between January 2014 and March 2015. The z-axis represents the number of targeted organizations. (Source: Dell SecureWorks)

In 2013, attackers began avoiding countries where international transactions are more difficult and require local intervention to launder money, shifting their focus to countries where institutions have weaker account security. As a result, CTU researchers observed a spike in attacks against Asian banks and institutions in 2014. Targets included traditional banks; institutions facilitating high-volume/high-value transactions, such as Automated Clearing House (ACH) or Single Euro Payments Area (SEPA) credit transfers; corporate bank accounts; and payroll systems.

Features

Although banking botnets have different features, sizes, and technical proficiency, they all focus on stealing financial information and using compromised systems for monetary gain. Many banking trojans steal email credentials from compromised systems and use them in spam campaigns to compromise more systems. Banking botnets became more widespread, resilient, and evasive in 2014. Dyre, Bugat v5, and Gozi/Vawtrak integrated multiple backup C2 solutions using proxy servers concealing real C2 servers, P2P networks, domain generation algorithms (DGAs), and anonymizing services such as Tor and I2P.

Man-in-the-browser (MITB) remains the most common and widely used attack technique in banking botnets, but Table 2 lists other features that botnets offer to attackers. Cybercriminals often combine features; for example, using MITB to hijack a web session, redirect and virtual networking computing (VNC) / backconnect features to control fraudulent transactions, screenshots and video captures to capture important information, and proxies to tunnel traffic and conceal C2 activity. Combining MITB attacks against browsers with social engineering attacks to compromise mobile devices also allows cybercriminals to circumvent security measures such as one-time passwords and two-factor authentication.

Feature	MITB	Redirect	VNC / Back connect	Screenshots	Video capture	Proxy	Certificate stealer	Status
Zeus	Y	Y	Y	Y	Plugin	Y	Y	Active
IceIX	Y	Y	Y	Y	Plugin	Y	Y	Active
Citadel	Y	Y	Y	Y	Plugin	Y	Y	Active
Gameover	Y	Y	Y	Y	N	Y	Y	Not Active
KINS	Y	Y	Y	Y	Plugin	Y	Y	Active
Shylock	Y	N	Y	N	Y	Y	Y	Not Active
Bugat v4 (Geodo)	Y	Y	Y	Y	N	N	Y	Active
Bugat v5 (Dridex)	Y	Y	Y	Y	N	Y	Y	Active
Gozi	Y	N	Y	Y	N	Y	Y	Active
Dyre	Y	Y	Plugin	Y	Y	Y	Y	Active
Ramnit	Y	Y	Y	Y	N	N	N	Not Active
Tinba	Y	Y	Y	Y	N	Y	N	Active
Hesperbot	Y	Y	Plugin	Plugin	Plugin	Plugin	Plugin	Active

Table 1
Feature list of banking botnets as of March 2015.

After attackers obtain a victim’s banking website credentials, they use a proxy server to connect to the victim’s computer via VNC and access the account directly. VNC bypasses some account protection mechanisms because the website recognizes the victim's web browser and allows actions such as transferring money. After accessing a victim’s account, attackers can transfer money into an account under their control. In many cases, the attackers move the stolen money through a series of victims’ accounts to make their activities difficult to trace.

Infection vectors

Banking trojans compromise systems via many different methods, including spam campaigns, downloader trojans, and drive-by download attacks using various exploit kits.

- Most of the trojans use port 80 (HTTP) or 443 (HTTPS) for communications between victims' systems and the C2 server. These ports are typically not blocked or monitored for outbound connections.
- In 2014, CTU researchers observed banking botnets using four primary downloaders: Pony, Upatre, Gamarue (also known as Andromeda), and Chanitor. These downloaders, which serve as first-stage droppers/installers, use sophisticated techniques to avoid detection when dropping malware on a compromised system.
- Between January 2014 and March 2015, CTU researchers observed the Magnitude, RIG, Nuclear, Cool, Styx, and Blackhole exploit kits distributing banking trojans.
- In 2014, Bugat v5 and Dyre introduced private spam mailers that run multiple spam campaigns and distribute new trojans every day.

Active banking botnets

The following banking botnets affected financial institutions around the world in 2014 and early 2015.

Dyre

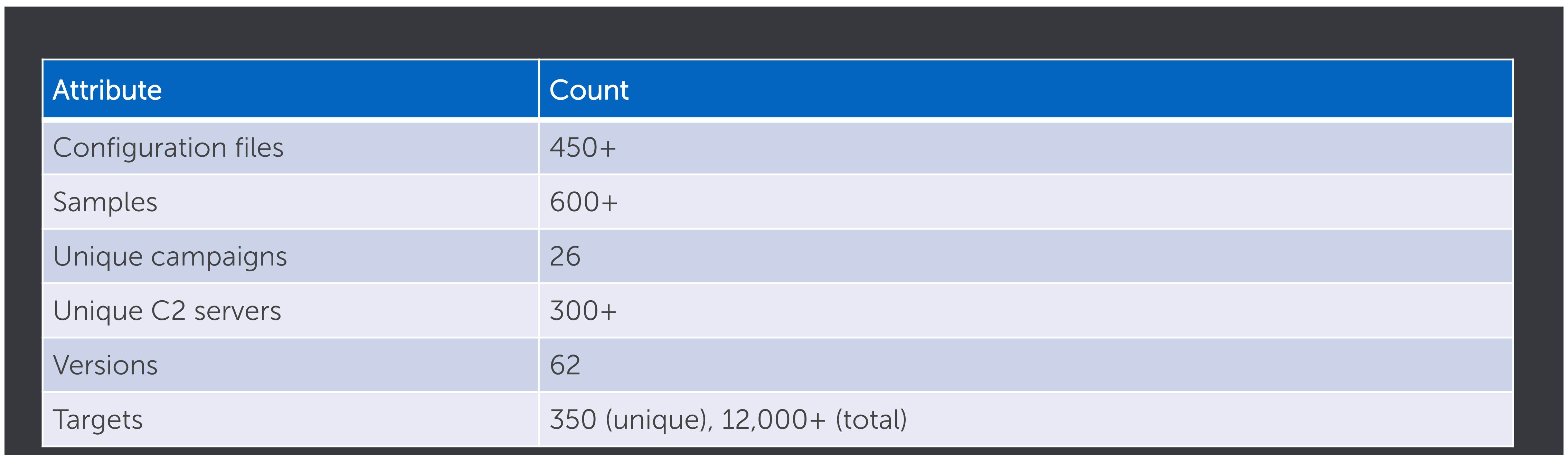
In early June 2014, CTU researchers discovered the Dyre banking trojan (also known as Dyreza, Dyzap, and Dyranges), which was being distributed by Cutwail botnet spam emails. Dyre was initially distributed via links to either Dropbox or Cubby file storage services, but it later began using the Upatre downloader trojan. Dyre has emerged from its primitive origins to become one of the most prominent banking trojans. Since its introduction, the CTU research team has identified 21 unique Dyre campaigns targeting more than 432 financial institutions around the world.

Dyre has incorporated refinements and new features in each iteration, evolving from only being able to intercept SSL traffic to having an advanced modular structure capable of using web fakes, dynamic web injects, and multiple options to maintain control of the botnet. Early Dyre versions could intercept SSL traffic and post it to a C2 server in clear text. The most recent version as of this publication uses SSL to encrypt all C2 communications. Dyre also introduced a custom algorithm and RSA cryptography to digitally sign configuration files and malware plugins, preventing data tampering. The malware is divided into two parts: the dropper and the main DLL module. The module, which is available for 32-bit and 64-bit Windows versions, hides the base configuration, RSA public key, and Campaign ID in its resource section.

Dyre uses a slightly different web inject engine than classic banking trojans. It hooks code into the Firefox, Chrome, and Internet Explorer web browsers to intercept all web-session data. The malware intercepts and sends the data to a drop server via an HTTP POST request. Dyre can dynamically manipulate banking website content.

Dyre hides its backend infrastructure behind a set of proxy servers that act as C2 servers. The CTU research team determined that most of the proxy servers are located in North America and Europe. The latest Dyre version as of this publication introduced two mechanisms to maintain control of the botnet if the proxy servers are unreachable: a DGA and a plugin that integrates with the I2P network. The DGA is seeded by the current data and generates 1,000 34-character domains per day. Dyre uses eight country code top-level domains (ccTLDs) in Asia and Pacific Islands: .cc, .ws, .to, .in, .hk, .cn, .tk, and .so.

Table 2 lists the statistics for Dyre samples and configuration files analyzed by CTU researchers.



Attribute	Count
Configuration files	450+
Samples	600+
Unique campaigns	26
Unique C2 servers	300+
Versions	62
Targets	350 (unique), 12,000+ (total)

Table 2

Dyre samples and configuration files analyzed by CTU researchers between its inception in 2014 and March 2015.

Gozi (Vawtrak and Neverquest)

First discovered and named by the CTU research team in 2007, Gozi (which includes the Vawtrak and Neverquest variants) offers very powerful capabilities and has an operational scope similar to Zeus. Traditionally, Gozi primarily spread through spam campaigns, redirecting victims to drive-by download exploit kits that installed Gozi on compromised systems. In September 2014, Gozi operators began using the Chanitor downloader, also known as Limpopo or Hancitor, to distribute malware. Chanitor downloads Gozi payloads through Tor.

Gozi is divided into two parts: a dropper module and the main DLL module. The dropper loads a DLL that initializes the main Gozi DLL module. After a system is compromised, Gozi connects to a predefined list of C2 servers and registers a bot. The C2 server responds with an encrypted configuration file that includes a list of banking websites and corresponding web inject scripts. When a victim attempts to log into one of the targeted sites, the trojan activates itself and steals the victim's credentials. Gozi can steal login credentials from FTP, SMTP, and POP applications. Attackers use FTP credentials to download other malware onto compromised websites or stage future Gozi attacks, and use stolen email credentials in future spam campaigns. Gozi can also harvest data from Google, Yahoo, Amazon AWS, Facebook, Twitter, and Skype. Attackers use these accounts to spread links to compromised websites to further spread Gozi and other malware.

Gozi's configuration defines targeted websites that belong to large international banks and popular online payment services around the world. In addition to these predefined sites, the malware can identify web pages containing specific keywords such as "balance," "checking account," and "account summary." Gozi collects the content of these pages to identify potential financial targets and to build web inject scripts.

Table 3 lists the statistics for Gozi samples and configuration files analyzed by CTU researchers.

Attribute	Count
C2 servers	400+
Configuration files	3,300+
Samples	2,800+
Campaigns	230+
Versions	10
Targets	108 (unique); 28,000+ (total)

Table 3

Gozi samples and configuration files analyzed by CTU researchers between January 2014 and March 2015.

Bugat (Bugat v5 (Dridex) and Geodo)

Initially positioned as a Zeus alternative, Bugat first appeared in January 2010 and has an aggressive versioning history. Each generation has a distinct message data structure and encryption scheme, and the malware reuses existing libraries and formats for greater flexibility and extensibility. The Bugat v5 (Dridex) and Geodo variants were introduced after the Gameover Zeus takedown, and CTU researchers have observed three versions of these variants: one containing hard-coded C2 servers in each sample, one containing a DGA, and one using a P2P network for its C2 communications.

Bugat grew significantly in 2014, moving from a centralized C2-based architecture to a P2P architecture. The malware, which has a downloader and a main DLL module, registers a bot to its C2 server after gathering basic system information from the victim's system, including serial number, computer name, version information, and a hash value of the user's security identity. Bugat uses an XML-based message architecture for its C2 communication. Its customized cryptographic system combines public-key cryptography (RSA) with symmetric-key cryptography (RC4), providing the confidentiality of non-symmetric encryption and the efficiency of symmetric encryption.

The Bugat variants capture form data from SSL pages, use web injects for HTML manipulation, modify local files, and steal credentials from web sessions. Similar to other popular banking trojans, Bugat hooks WinINet and NSPR functions to leverage the malware's data stealing and web injection capability against web browsers like Internet Explorer and Firefox. Bugat v5 differs from previous variants, particularly in its modular architecture and use of a hybrid P2P network to mask its backend infrastructure and complicate takedown attempts. Bugat v5 has four modules: a loader module that downloads the core module and initial P2P node list, a core module that harvests credentials through MITB attacks and downloads other modules, a virtual network computing (VNC) module that allows the attacker to remotely view and control a victim's computer, and a backconnect module that allows the attacker to tunnel network traffic through a victim's computer.

Similar to Gameover Zeus and Gozi Neverquest, Bugat v5 operates with an affiliate model. It is partitioned into sub-botnets, and each affiliate has access to its own subset of bots. The Bugat v5 C2 servers multiplex bot requests according to a botnet value contained in each request. The malware's P2P network leverages existing bots to relay traffic between bots and the criminal infrastructure. Bugat v5 bots that have a public IP address and are not behind a NAT or firewall act as nodes in a P2P network. These nodes attempt to listen on TCP ports, and the P2P messages are encapsulated using the HTTP POST format.

The Bugat v5 P2P network is a hybrid network. Rather than nodes behaving autonomously and exchanging peer lists, configuration files, and binary updates with other peers, they tunnel nearly everything to the backend infrastructure. Bots that perform node actions receive a special information packet that contains the location of an admin node (i.e., an upstream proxy).

Table 4 lists the statistics for Bugat samples and configuration files analyzed by CTU researchers.

Attribute	Count
C2 servers	100+
Configuration files	1,600+
Samples	1,000+
Networks	9
Versions	19
Targets	97 (unique); 20,000+ (total)

Table 4
Bugat samples and configuration files analyzed by CTU researchers between January 2014 and March 2015.

Gameover (P2P) Zeus

Gameover Zeus, which emerged in July 2011 shortly after the leak of Zeus source code, performs operations ranging from simple credential stealing to advanced methods like hijacking victims' bank accounts in real time. Although it is based on Zeus source code, Gameover Zeus introduced a decentralized control system with its P2P architecture. A BotID uniquely identifies each bot within the P2P network, and the proxy nodes act as designated relay points for botnet operators to send commands and receive stolen information.

Gameover Zeus hides its criminal infrastructure behind a robust network architecture of TCP and UDP-based P2P communication. TCP is used to transmit malware-specific data, such as configuration and executable updates, and UDP is primarily used to maintain the P2P infrastructure, such as sharing lists of known peers. The botnet used the P2P protocol to receive configuration and binary updates from other peers; stolen data and instructions were relayed through a peer using RSA-2048 and RC4 encryption. A DGA dynamically generated a pool of 1,000 domain names each day and served as a failsafe mechanism if the P2P infrastructure was compromised or unreachable.

In early June 2014, security researchers exploited design flaws in the Gameover Zeus P2P network during Operation Tovar. They disrupted the criminal infrastructure by manipulating the P2P peer list and redirecting traffic to nodes under the researchers' control.

Table 5 lists the statistics for Gameover (P2P) Zeus configuration files analyzed by CTU researchers.

Attribute	Count
Configuration files	189
Samples	24,000+
Unique bot IDs	160,000-170,000 (per day)
Unique IP addresses	220,000-240,000 (per day)
Versions	3
Targets	495 (unique); 23,800 (total)

Table 5

Gameover Zeus P2P samples and configuration files analyzed by CTU researchers between January 2014 and the Operation Tovar takedown in June 2014.

DGA-based Gameover Zeus

One month after Operation Tovar, researchers identified a new Gameover Zeus variant that shares the basic Gameover Zeus code but omitted P2P functionality. Instead, it only uses DGA for its backend and C2 communications. The malware generates domain names based on the current date and a predefined magic value.

The CTU research team has identified three versions of this Gameover Zeus family. Two of the variants generate 10,000 new domains every day, and the other generates 1,000 new domains every day. The botnet operators use fast-flux to switch IP addresses associated with domains, allowing them to bypass IP-based blacklists. This DGA-based Gameover Zeus variant follows the same post-infection operations as P2P Gameover Zeus. It exhibits the same behavior, uses the same PCRE-based configuration file structure, and employs the same hooking techniques to gain direct access to raw HTTP data.

Table 6 lists the statistics for DGA-based Gameover Zeus configuration files analyzed by CTU researchers.

Attribute	Count
Configuration files	3
Samples	89
Unique bot IDs	8
Versions	3
Targets	189

Table 6

DGA-based Gameover Zeus samples and configuration files analyzed by CTU researchers between January 2014 and the Operation Tovar takedown in June 2014.

Zeus

The Zeus banking trojan (originally called PRG or Zbot) was first discovered by the CTU research team in 2007. Since the Zeus source code was leaked in 2011, almost all banking trojans have incorporated Zeus features. As of this publication, Zeus is still very effective, compromising thousands of systems and resulting in the theft of hundreds of millions of dollars. The Zeus toolkit contains three parts: a builder that allows the attacker to build the trojan, the Trojan horse malware that modifies a compromised computer and steals information, and a C2 web panel that monitors and controls the trojan and stores stolen data.

The Zeus architecture is simple. Each bot is programmed to connect to a specific C2 server, and the dynamic configuration allows the botnet operators to update the C2 server location. Cybercriminals can rent individual Zeus servers and orchestrate their own banking campaigns. Due to the widespread availability of the Zeus control panel code, CTU researchers have observed it used on many compromised servers.

The statistics listed in Table 7 confirm that Zeus was an active and effective banking botnet in 2014.

Attribute	Count
C2 servers	1,000+
Configuration files	1,300+
Samples	8,000+
Encryption keys	550+
Versions	11
Targets	740 (unique); 163,812 (total)

Table 7

Zeus samples and configurations analyzed by CTU researchers between January 2014 and March 2015.

IceIX (Ice9)

The IceIX credential theft trojan, which is based on the Zeus source code, does not appear to offer any unique functionality. The minor differences in IceIX are the inclusion of the IceIX version rather than Zeus version number in configuration files, the use of a slightly modified RC4 algorithm instead of the standard RC4, and a custom HTTP POST request to download IceIX's dynamic configuration.

Table 8 lists IceIX statistics. The malware is still active and effective as of this publication, but activity is steadily decreasing.

Attribute	Count
C2 servers	380+
Configuration files	500+
Samples	650+
Encryption keys	92
Versions	6
Targets	1,017 (unique); 31,200 (total)

Table 8

IceIX samples and configuration files analyzed by CTU researchers between January 2014 and March 2015.

Citadel

The Citadel banking trojan is based on the leaked Zeus source code and, like Zeus and IceIX, is composed of three parts. Citadel introduced major improvements over Zeus, such as revised cryptography, sandbox detection, DDoS capability, command execution, and aggressive DNS filtering. Citadel also added new functionality, including custom AES encryption of configuration files, a custom communication protocol over HTTP, blocking/redirecting of security sites on victims' systems, and the ability to record videos of activities on compromised systems. Citadel's development appears to have stalled; the latest version observed by CTU researchers as of this publication is from April 2014.

Like Zeus and IceIX, the Citadel trojan is programmed to connect to a preconfigured list of C2 servers that issue commands. Attackers can update the C2 server options with a dynamic configuration file, and cybercriminals can rent individual servers to orchestrate their campaigns. Citadel uses the API hooking technique in its compromises, stealing and logging functionality from victims' systems.

The CTU research team observed a Citadel 3.1 variant on the Internet in early 2014. This variant introduced the ability to spread via external devices such as USB by taking advantage of the "autorun.inf" functionality. It also introduced a "port scan" command and added a new encryption layer for both communication and the configuration file.

The statistics in Table 9 show that Citadel is active and effective as of this publication despite its lack of updates.

Attribute	Count
C2 servers	900+
Configuration files	2,200+
Samples	22,000+
Encryption keys	300+
Versions	5
Targets	1,170 (unique); 137,000 (total)

Table 9

Citadel samples and configuration files analyzed by CTU researchers between January 2014 and March 2015.

KINS

First advertised in July 2013, KINS (also known as VMZeus and Zberp) is also based on the leaked Zeus source code, and its toolkit also has three parts. KINS introduced major improvements such as revised cryptography, concealment of configuration files inside digital image files, sandbox detection, reporting of installed security product information, and command execution.

Each KINS sample is embedded with information such as web inject download locations and encryption/decryption keys for the dynamic configuration. To avoid malware trackers, KINS includes a build-time generated virtual language interpreter that can be built with 16 DWORD registers. KINS also includes VNC functionality that allows connections through a compromised computer, so attackers can access a victim’s bank account while appearing to originate from the victim’s IP address or computer. This method circumvents IP address or device fingerprint-based fraud detection mechanisms.

KINS introduced a “config steganography” feature that allows the malware to disguise its configuration in digital image formats. It encrypts its dynamic configuration using XOR and RC4/RC6 crypto modules, Base64-encodes the encrypted configuration, and appends the result to the end of a legitimate digital image. CTU researchers also observed two KINS versions using a novel “invisible persistence” feature. The malware deletes its startup registry key when Windows starts and sets it again while Windows shuts down, evading antivirus software that scans for malware during system boot.

Table 10 lists the statistics for the KINS samples and configurations analyzed by CTU researchers.

Attribute	Count
C2 servers	511
Configuration files	350
Samples	6,000+
Encryption keys	893
Versions	8
Targets	480 (unique); 2,790 (total)

Table 10
KINS samples and configuration files analyzed by CTU researchers between January 2014 and March 2015.

Shylock

Shylock (also known as Caphaw) was first discovered in the second half of 2011. It was not as widespread as other popular banking trojans and was never openly advertised for sale. CTU researchers speculated that the malware was used by a single group and was never sold separately. It was distributed via spam campaigns and drive-by download attacks through different exploit kits, as well as through local shares and removable drives.

Shylock has many features included in popular banking trojans. It hooks into web browser processes and monitors activity for websites of interest. It can inject HTML and JavaScript code into specific web pages, steal or delete HTML and Flash cookies, take screenshots or record videos of specific web pages, and upload system attributes and stolen information to C2 servers. Shylock uses HTTPS to encrypt C2 communications for receiving and uploading information, and C2 URLs are hard-coded in the binary. Shylock uses API hooking to insert itself into a process's program flow. The malware uses Windows named pipes to send messages between the injected processes ("slave" instances) and a "master" instance that typically runs in explorer.exe. The slave instances use these messages to send stolen data to the master instance for uploading to the C2 server, or to retrieve configuration information from the master instance.

Shylock uses three server types to allow attackers to perform transactions: a C2 server to determine targets, a VNC server to remotely log into compromised systems, and a backconnect server to tunnel traffic through the compromised systems. It also uses web inject servers to intercept traffic during MITB attacks. Shylock implements a robust plugin-based architecture and divides its functionality into small, separate modules.

In early July 2014, a global law enforcement operation seized C2 servers and botnet-related domains that significantly affected the Shylock infrastructure.

Table 11 lists the statistics for Shylock samples and configuration files analyzed by CTU researchers.

Attribute	Count
C2 servers	53
Configuration files	1,755
Samples	1,652
Botnets	9
Versions	6
Targets	66 (unique); 890 (total)

Table 11
Shylock samples and configuration files analyzed by CTU researchers between January 2014 and the July 2014 takedown.

Tinba

First discovered in 2012, the initial Tinba (also known as Tiny Banker and Zusy) versions were approximately 20KB in size and typically targeted Turkish banks. After the source code was leaked in 2014, a sophisticated version of Tinba emerged and is being used to attack banks around the world.

Tinba uses API hooking techniques to gain control of compromised systems. It stages the compromise via the legitimate Winver.exe Windows process and then uses explorer.exe or svchost.exe to perform malicious operations. Tinba hooks WinINet APIs to perform browser injection and interception, and lowers browser security settings to perform browser injection.

Each Tinba sample uses an embedded static configuration as web injects if its C2 server is unreachable. Tinba uses RC4 to encrypt its C2 communication and DGA as a fallback mechanism to phone home if the C2 communication fails. Tinba signs its commands with a public key to guarantee they originated from a legitimate bot operator. Tinba uses the same configuration file structure as Spyeye and Zeus, placing 'G' after a target URL to indicate a trigger on GET requests, a 'P' after a target URL to indicate a trigger on POST requests, and a '!' before a URL to exclude it from target lists.

Table 12 lists the statistics for Tinba samples and configuration files analyzed by CTU researchers.

Attribute	Count
Configuration files	150+
Samples	650+
Versions	3
Targets	90 (unique); 890 (total)

Table 12
Tinba samples and configuration files analyzed by CTU researchers between January 2014 and March 2015.

Ramnit

Ramnit was first discovered in early 2010, and a sophisticated variant released in late 2011 evolved into a banking trojan. Ramnit can monitor web sessions and steal banking credentials from compromised systems. Although Ramnit initially targeted Southeast Asia, it slowly expanded its target base to victims across the world. The latest Ramnit version as of this publication primarily focused on UK and European banks.

The malware consists of several components divided into small modules/plugins, including a dropper, a Zeus-like MITB bundle, an FTP grabber, a VNC module, a form and cookie grabber, and an anti-antivirus module. It uses a configuration file similar to Spyeye and Zeus and spreads via multiple infection vectors, including network and removable drives, malicious files, exploit kits, social media, and public FTP services. It injects its DLL module into newly created instances of explorer.exe or svchost.exe and periodically communicates with its C2 server, which is determined by a DGA. Ramnit can receive and execute commands on behalf of the attacker and can request additional modules, which are RC4-encrypted.

In early 2015, a joint effort between law enforcement and industry partners resulted in the seizure of servers and other infrastructure owned by the group behind Ramnit.

Table 13 lists the statistics for Ramnit samples and configuration files analyzed by CTU researchers.

Attribute	Count
Configuration files	187
Samples	480
Botnets	15
Versions	2
Targets	329 (unique); 13,980 (total)

Table 13

Ramnit samples and configuration files analyzed by CTU researchers between January 2014 and the February 2015 takedown.

Hesperbot

Hesperbot, first discovered in mid-2013, contains common banking malware functionality. Its initial campaigns targeted the Czech Republic and Turkey, but the target base expanded in 2014 to include banks and financial institutions around the world.

The malware includes x86 and x64 versions of various modules: a dropper module that injects the core module into legitimate Windows processes; a core module that handles C2 communications, registers a bot with its C2 server, and downloads and launches other modules; a keylogger module that intercepts keystrokes on a compromised system; a VNC and SOCKS module that allows attackers to remotely connect to and control a compromised system; a proxy module that sets up a local proxy to intercept SSL traffic and hook certificate verification APIs; and an injection module that captures screenshots and videos, grabs forms, and handles web injection in live web sessions. It uses DGA as a fallback mechanism to phone home if C2 communications fail. Hesperbot also uses a mobile component to bypass two-factor authentication or one-time password authentication schemes. This component targets Symbian, Blackberry, and Android platforms, supporting a broad range of devices.

Table 14 lists the statistics for Hesperbot samples and configuration files analyzed by CTU researchers.

Attribute	Count
Configuration files	184
Samples	228
Variants	2
Botnets	12
Targets	83 (unique); 2,280 (total)

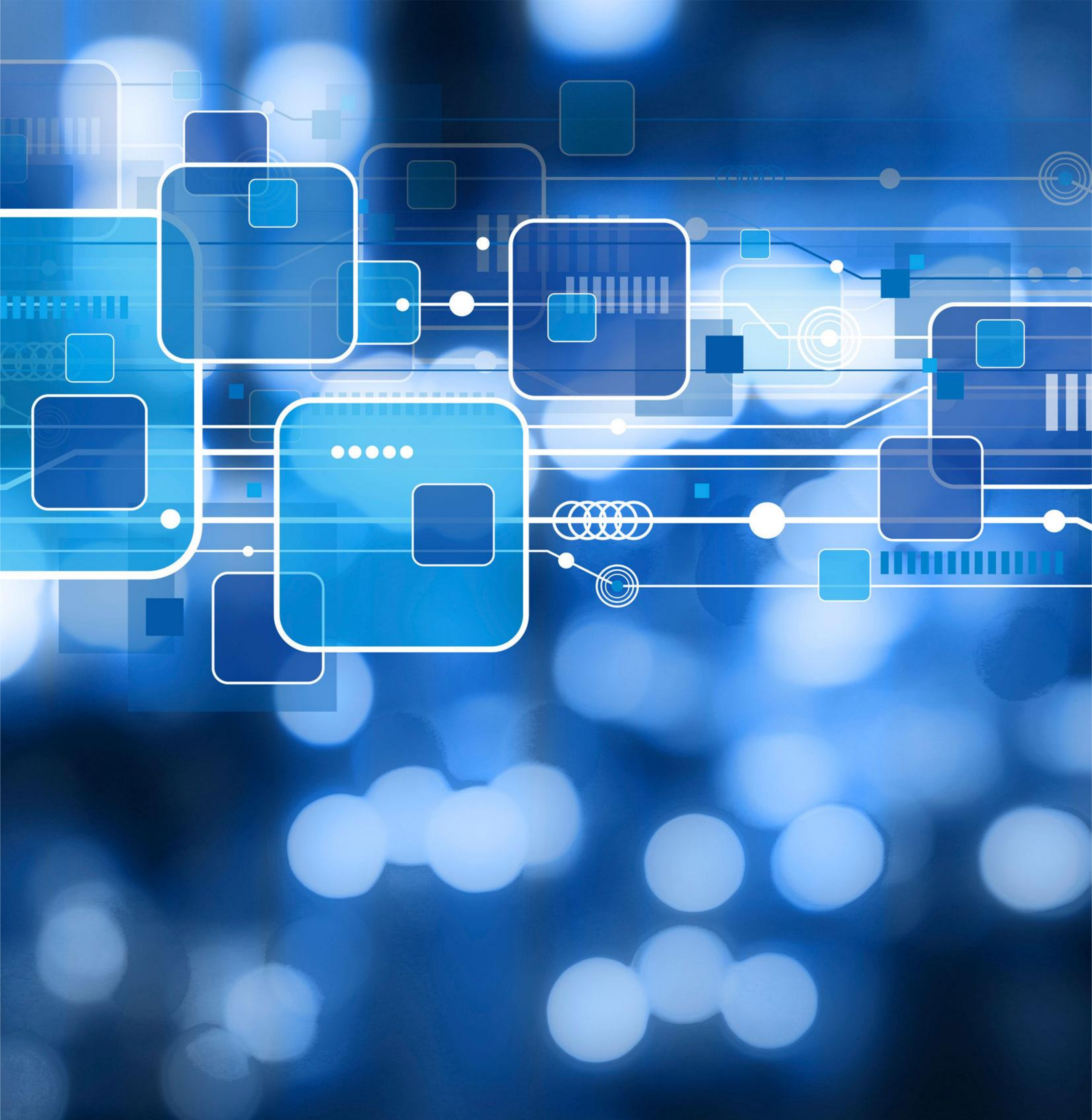
Table 14

Hesperbot samples and configuration files analyzed by CTU researchers between January 2014 and March 2015.

Conclusion

Takedowns and arrests temporarily reduced banking botnet activity in 2014 and early 2015. Although the operations had some success, the introduction of Dyre and Gameover Zeus DGA shortly after the Gameover Zeus and Shylock takedowns reflected the determination of attackers targeting the financial vertical. Cybercriminals also leverage well-organized service industry and online marketing tools to promote and sell their services, as well as third-party services to circumvent security measures. In addition, attackers continually expand their operations to new markets and locations where they can apply existing techniques.

Although CTU researchers did not observe much innovation in fraud techniques in 2014 and early 2015, traditional solutions to protect against threats prove ineffective against modern banking trojans. The CTU research team recommends that clients conduct online banking and financial transactions on isolated workstations that are not used for web browsing, reading email, and other activities that could increase the risk of infection. The best defense for financial institutions is a unified web security solution with real-time content inspection of every packet of incoming and outgoing web content. Automated attack detection requires collecting, combining, and automatically analyzing data to extract relevant information and apply security countermeasures. Combining this data with intelligence on known botnets will help enlarge the knowledgebase for identifying attacks and selecting appropriate attack mitigation tools.



SecureWorks Banking Botnets Persist Despite Takedowns | APRIL 2015

About Dell SecureWorks Dell SecureWorks is a market-leading provider of world-class information security services with more than 3,800 clients in 70+ countries. Organizations of all sizes rely on Dell SecureWorks to protect their assets, improve their compliance and reduce their costs. Our combination of award-winning security expertise and client support makes Dell SecureWorks the premier provider of information security services.

Dell SecureWorks uses cyber threat intelligence to provide predictive, continuous and responsive protection for thousands of organizations worldwide. Enriched by intelligence from our Counter Threat Unit research team, Dell SecureWorks' Information Security Services help organizations proactively fortify defenses, continuously detect and stop cyber attacks, and recover faster from security breaches. For more information, visit: http://www.secureworks.com.

For more information, phone 877.838.7947 to speak to a Dell SecureWorks security specialist.