Not so secure!

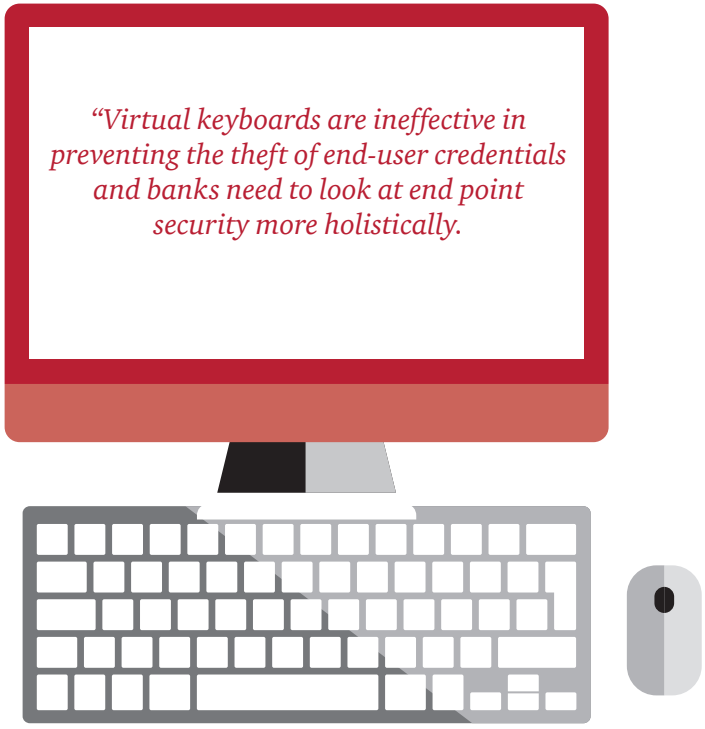
Research report on end-point compromises and the risk to Internet banking

Special research report.
August 2016

pwc

Internet Banking has grown by leaps and bounds in India. India had over 400 million Internet users by the beginning of 2016 as per a report by IAMAI and IMRB. This is the highest number of Internet banking users in any country.

While having a high number of Internet banking users indicates healthy adoption of IT and reduced costs for banks, it also means that this population is vulnerable to cyberthreats which can put their hard-earned money at risk. One of the biggest addressable risks which users face is man-in-the-browser and end-point compromise attacks.



End-point compromise is an attack in which the end user's computer and, in many cases these days, a browser is compromised using a simple program—generally a browser plugin or browser extension. The program is designed to steal the Internet banking credentials of the end users. Our security research team looked at the Internet banking websites of 26 banks in India. We found that over 21 banks have inadequate mechanisms to detect or prevent attacks which could compromise the end user credentials. This finding has significant implications as the end user is generally gullible, and many users can be made to download a browser extension/plugin which is capable of capturing banking credentials. Mechanisms which are currently prevalent, such as the virtual keyboard, which were long considered a protection against such attacks, were found to be inadequate to prevent these attacks. Most banks today employ a second-factor authentication, generally an SMS OTP for transacting. To illegally transfer funds from an account, an end-point compromise attack needs to work in conjunction with the second attack which steals the OTP from an Android device. We were able to successfully demonstrate both attacks working together using a specially created mobile malware.

Over 200 million Internet banking users in India do not have even higher secondary education. Even those who do can inadvertently be persuaded to download a plugin which can compromise their accounts. Even without the second-level attack of stealing an OTP, the attack puts the data of millions of users at risk. Considering the trust an end user puts in the Internet banking infrastructure, it is essential that banks look at security more holistically. What makes this finding serious is the enormity of the scale at which such attacks can be executed.

The remainder of this document discusses the vulnerability in detail and some countermeasures which banks can take to mitigate this vulnerability. While researching this vulnerability, we did not carry out any intrusive activity. The vulnerabilities were discovered by monitoring the publicly available information.

Detailed commentary

End-point compromise: The mechanisms

Over the years, financial Trojans have emerged as the key threat to financial organisations. In 2015, 36% of the attacks were classified as malware (which is higher than the percentage of web application attacks). This number was a mere 4% in 2014 . In the security circle, these tools are known as 'crimeware'. Typically, most Trojans which operate at the operating system layer will be detected by an anti-malware solution. However, a crimeware operating as a browser plugin is not detected by most antiviruses or anti-malware solutions. Moreover, these can easily masquerade as a productivity enhancing add-in.

Since Internet banking transactions take place inside a browser, a browser plugin is the easiest tool that can be created for end-point compromise. This browser plugin can be served to the end user though a phishing campaign, through free download websites or by giving the browser plugin a name that is very similar to popular browser plugins such as 'Google Translate' or 'Adobe Reader'. This attack can be executed in a slow and stealthy mode, with the plugins delivering real functionality and gaining popularity over time and thus to infecting a large population.

Typical attack vectors for browser-based compromise will be:

1. **DOM/HTML manipulation**

   - The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. When a web page is loaded, the browser creates a DOM of the page. This object model can be freely manipulated by JavaScript in order to perform various malicious activities such as:
   - HTML element manipulation: It can be used in order to add new HTML elements (addition of extra normal fields such as card number or CVV number), remove elements (warning messages about phishing) or even change elements (change content of the web application).
   - HTML attribute manipulation: Attributes are additional information set for a particular element. By manipulating these attributes, new properties can be set (adding custom functions), existing properties can be changed (Setting Password field as text), or existing properties can be deleted (Removal of crucial encryption functions)
   - HTML Event Manipulation: Events are generated when a particular activity such as loading, clicking etc. occur. An attacked can add a new event (Custom function on form submission), change an existing event listener (change core functions which occur on form submission) or delete event listeners (remove functions which change form parameter on form submission).

2. **Key loggers and password stealers**

   - Key loggers have long been used to compromise a user's credentials. Modern key loggers use browser hooks which start recording keystrokes if certain conditions are met (active in the bank's log-in page). Most modern banks use virtual keyboards to combat this challenge. However, these can be overcome by other techniques as below.
   - Password stealers, on the other hand, simply get the value of the input field (password stolen in plain text despite using a virtual keyboard).

3. **Ajax data theft**

   - Asynchronous JavaScript and XML (Ajax) is a set of web development techniques which run on the client side to send and retrieve data from a server asynchronously (running in the background and not interfering with the display and behaviour of the existing page). Ajax can be used to send malicious data (value of username and password fields, clipboard content, etc.) to any server, which can then be used as a drop zone.

4. **Form grabbing**

5. Form grabbing is an extension of DOM Manipulation, which works by retrieving the authorisation and log-in credentials from a web data form before it is passed over the Internet to a secure server. This allows the attacker to avoid HTTPS encryption.

   Most modern banks encrypt the password before sending data. However, broken cryptography can lead to the password being reused or predicted.

6. **Form redirection**
   - Form redirection is a simple method wherein the action attribute of the form element is changed to a malicious domain. This causes the entire form to be dumped in the attacker's malicious domain.

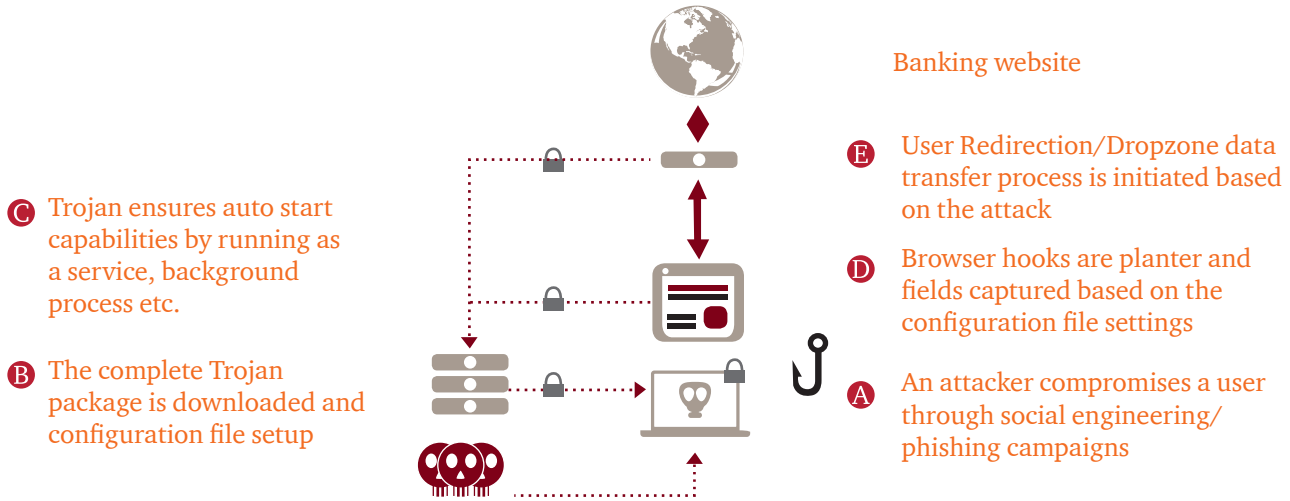
7. **Automatic transaction**
   - For banking applications, a transaction is the core component of the web application. Automated transactions generated through DOM manipulation can cause unauthorised or illegal transactions to occur.
   - Modern automated transactions even include components which repeatedly manipulate the visible balance in the web application so that the end user is not suspicious of such a transaction taking place.

Exploitation methodology

The exploitation method has a complex design so as to bypass security mechanisms. The exploit contains the following crucial components:

1. Command and control (C2) server: They are used by attackers to maintain communications with compromised systems within a target network.
2. Configuration file: Once a malware is installed on a victim's machine, it fetches a configuration file from one of its command and control servers. The file instructs the malware about the websites and applications to target, which information to steal and how to steal it. This information is encrypted and is usually hard to capture.
3. Dropzone: A dropzone is a publicly writable directory on an Internet server that serves as an exchange point for malware data. The malware running on a compromised machine sends all stolen data/credentials to the dropzone, where the attacker can pick them up and start to abuse them.

The overall working of a basic exploit would be as follows:



Source: PwC analysis

Countermeasures

There are several methods a bank may apply to counter this threat. One of the easiest methods which makes this attack difficult to conduct is the forced used of a new browser window which does not communicate with the originating browser. However, care needs to be taken that the new browser window cannot be accessed directly through a static URL, which may defeat the purpose. Organisations may also look to implement measures such as checks for source webpage integrity with session termination or implement keystroke encryption. There also exist off the shelf solutions which provide protection mechanisms against such attacks.

Conclusion

The more we understand about security, the less we seem to know. End-user compromise and targeted attacks on customers are not just about end-user awareness. There is a lot that banks and corporates could do.

Attacks such as these have the potential to compromise multiple users and target high net worth individuals. While we focused on banking websites, the attacks mentioned herein are equally potent threats for other financial transactions, websites hosting corporate information, etc. It is time that corporates analyse how end-user or end-point compromises can affect their overall credibility and trust.

Annexure I: Our testing methodology

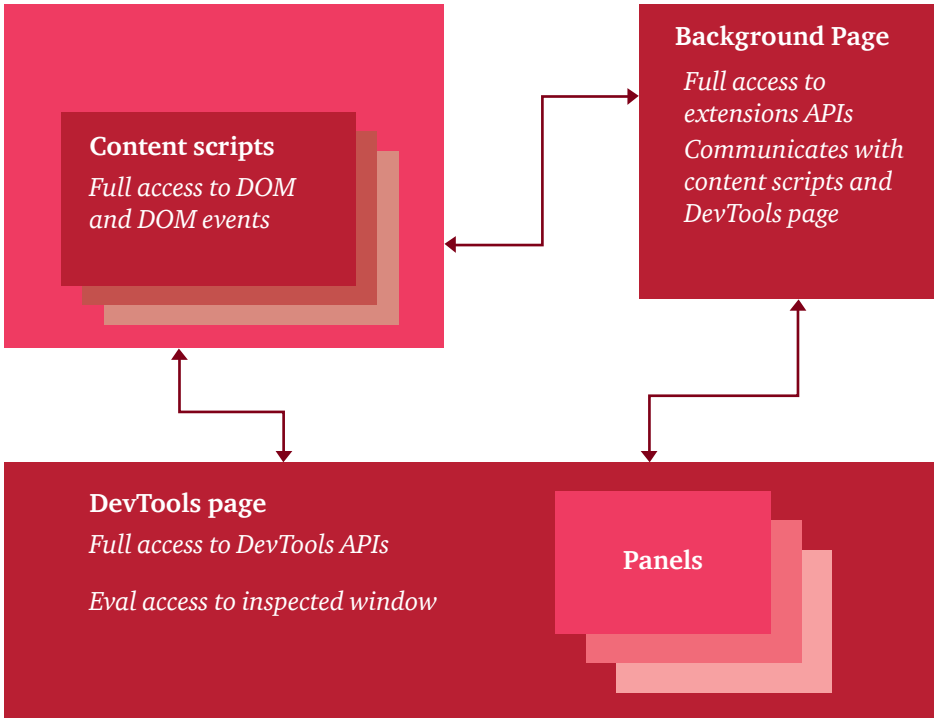
1 The architecture of a Chrome extension allowed for scripts to be run on the content window directly and for data to be sent via HTTP (even if the website has an HTTPS connection) via the extension.

2 Alternatively, the possible exploit options included coding a malware capable of running on the background of web pages, injecting scripts via user script managers (Tamper Monkey, Grease Monkey etc.), API hooking etc.

3 A mobile malware was written for Android that was capable of grabbing an OTP and sending it to the dropzone in real time, while preventing a notification from popping up and simultaneously deleting the message from the system. This app was concealed under an open source messaging app so as for efficient social engineering.

4 A dropzone was also configured so as to receive real-time inputs from the extension and measure the security hurdles in creating a dropzone without detection.

5 The dropzone was configured to receive inputs from the different vulnerability checks directly from the Chrome extension, inputs from a phishing page brought up locally via the extension, and inputs from a mobile malware capable of testing OTP security.



Source: developer.chrome.com

For more information, contact:

Sivarama Krishnan
Leader, Cyber Security, PwC
+91 9650788787
sivarama.krishnan@in.pwc.com

Sangram Gayal
Director, PwC
+91 9819197716
sangram.gayal@in.pwc.com

ME/September2016-7440