2011

Forensic recovery and analysis of the artefacts of crimeware toolkits

Murray Brand
Edith Cowan University

DOI: 10.4225/75/57b2b94e40ce8
Originally published in the Proceedings of the 9th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, 5th -7th December 2011

FORENSIC RECOVERY AND ANALYSIS OF THE ARTEFACTS OF CRIMEWARE TOOLKITS

Murray Brand
secau - Security Research Centre, School of Computer and Security Science
Edith Cowan University, Perth, Western Australia
m.brand@ecu.edu.au

Abstract

The total cost of cybercrime has been estimated to exceed US$388 billion annually. The availability of crimeware toolkits has lowered the bar for entry to the world of cybercrime. With very little technical knowledge required, cybercriminals can create, deploy and harvest financial data using banking trojans though a point and click graphical user interface that can cost less than US$1000. Technical support is also available for a fee, including technical infrastructure and servers to store harvested data. Fraudsters employing crimeware toolkits have been reported to have stolen US$3.2 million dollars in as little as six months. This paper presents preliminary research that has been conducted to forensically recover and analyse artefacts from the process of using crimeware toolkits from the file system and memory of systems that have been potentially engaged in such banking trojan authoring activities. Construction of a banking trojan using a crimeware toolkit follows a process that typically requires a set of configuration files and a small suite of program tools within the toolkit. Artefacts can be recovered from the process that could potentially be presented for admission as evidence in a court of law. Artefacts from the toolkits vary, as does the versions and variants of available toolkits. This paper proposes further research to construct a library of baseline artefacts to assist in the reconstruction of events to assist the forensic analyst in determining the provenance of any particular banking trojan.

Keywords

Digital forensics, crimeware, ZeuS, Spy Eye, Pinch, Carberp, cybercrime, banking trojan, botnet.

INTRODUCTION

A recent study estimates the cost of cybercrime worldwide to exceed US$388 billion dollars annually which exceeds the postulated US$288 billion cost of the global black market of illicit drugs such as marijuana, cocaine and heroin combined. The report also claims that more than two thirds of adults who go online have been a victim of cybercrime in their lifetime (Norton, 2011). Trend Micro (2011a, 2011b) reports that the top 10 cybercrime targets of the crimeware toolkits include online users of PayPal, eBay, Yahoo!, Facebook, Pharmacy Express, HSBC Bank, ANZ Bank, Lloyds TSB Bank, Banco Santander Bank and Western Union Bank.

Malicious software developers create, market and support crimeware toolkits that can create malicious software (malware) which can be used for the theft of financial transaction data such as bank account information, credit card data, authentication credentials and personal identities. Such targeted malware is typically referred to as a "banking trojan". The command and control infrastructure for the malware is in the form of a botnet which may include thousands of infected machines from which financial and personal data can be harvested. Instances of crimeware toolkits such as *ZeuS*, *Spy Eye*, *Pinch* and *Carberp* employ graphical user interfaces such that the cybercriminal needs very little technical knowledge to create a banking trojan (Trend Micro, 2011a, 2011b). A report by Trend Micro (2011c) reveals that an investigation into a certain cybercriminal's activities who employed such a toolkit to have stolen over US$3.2 million in six months from a botnet that employed over 25,000 systems, predominately located in the United States.

A generic, potential lifecycle for a banking Trojan built by a crimeware toolkit is depicted in Figure 1. The diagram shows that the toolkit is purchased from an online cybercrime site or obtained from some other means such as transfer from an external drive, email attachment or some other source. The toolkit is then installed on a system. The requisite files and tools may be on the single system or distributed across mounted volumes or network shares.

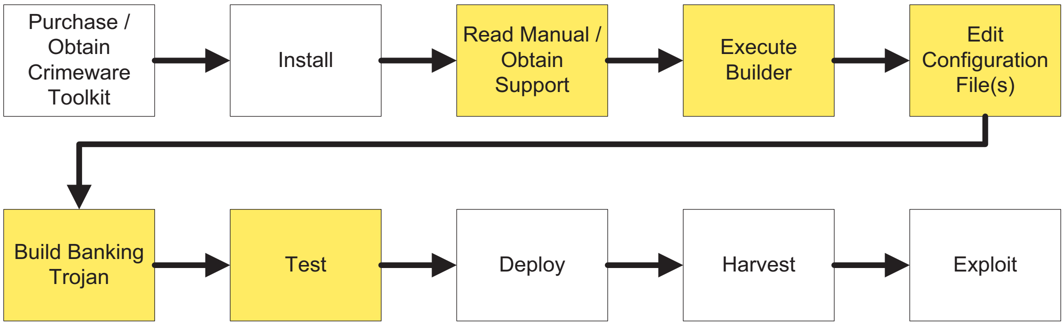
Purchase / Obtain Crimeware Toolkit
Install
Read Manual / Obtain Support
Execute Builder
Edit Configuration File(s)
Build Banking Trojan
Test
Deploy
Harvest
Exploit

Figure 1 Potential Crimeware Toolkit Usage Life Cycle

The crimeware toolkits examined for the purpose of this research all contained user's manuals and it is understood that online support is also available for a fee. The heart of the toolkit is a graphical user interface based "Builder" program that invokes a small number of supporting programs to build the resulting banking trojan and this process also requires the editing of a small number of configuration files. Once built, the banking trojan is likely to be tested in some manner, even if it is just to ensure that the resultant hash does not register as a known hash or virus signature. Once tested, it is envisaged that the trojan is deployed. The banking trojan can be deployed via a variety of mechanisms including spammed email that contains a link to a hijacked web site that uses *iframes* running malicious *javascript*. The malicious *javascript* can exploit vulnerabilities in the browser of the user which can then execute code on the target computer to conduct a variety of nefarious activities, including disabling security software and can download additional malware to the victim. Once compromised, the infected computer can be updated with enhanced malware at the discretion of the person in command of the botnet, typically referred to as the bot herder. The bot herder can then operate the botnet system from a distance, through a layered, hierarchical command and control system. Information can then be harvested and exploited in some predetermined measure.

The purpose of this paper is to present preliminary research that could be suitable for investigating a computer system suspected of having being used to author a banking trojan using a crimeware toolkit. The intention is to present artefacts of the authoring process as evidence suitable for admission to a court of law. This intention assumes that the sources of evidence accessible from the suspect machine have been acquired in a legal and forensically sound manner. Only the analysis phase of the investigation is discussed in this paper. This particular area of interest is depicted in the life cycle diagram of Figure 1 between reading the user manual to testing the resultant trojan and is highlighted in the figure. This paper does not investigate how the toolkit was originally acquired, nor does it investigate how the banking trojan was deployed, nor how the botnet was controlled, nor how information was harvested or exploited from the deployed banking trojan. These lines of investigation are left for future lines of research. The highlighted components in the figure emphasize the processes that are highly likely to have been conducted on one system. The components of the diagram that are not highlighted may have been conducted on the one system, but need not to have been.

Understanding the life cycle for any particular crimeware toolkit version or variant could assist an investigation. It can lead to the reconstruction of events, which can be represented in a timeline, which can be corroborated and supported with the artefacts from following the building process. Discovery of the artefacts from the combined acts of editing configuration files, running the builder program (which in turn runs subservient tools which leave artefacts) and accessing a user's manual may support the two essential elements required for a case. That is, *actus reus* (latin for guilty act) and *mens rea* (latin for guilty mind) which may be used to prove that the accused committed the prohibited act and possessed the culpable mental state (Shinder, 2002). Additional evidence such as the deployment of the trojan, the harvesting and exploitation of the resultant information from the trojan would likely provide additional, supporting elements to the case.

CRIMEWARE

Crimeware Toolkit Capabilities

Crimeware toolkits that include *ZeuS*, *Spy Eye*, *Pinch*, *Carberp* and *Bugat*, predominately operate under Microsoft Windows systems, but may also target alternative platforms such as mobile devices (S21sec, 2010). Malware continues to evolve, it is becoming more stealthy, increasingly targeted and incorporating additional anti-analysis techniques (Brand, 2010). As an example, Barrett (2011) lists features of *Spy Eye* to include a ring 3 rootkit which means it can hide registry and file entries from a limited privileges account. It can hook the supported web browsers such as Internet Explorer, Firefox and Maxthon, and then inject code into the browser. It can intercept and control traffic by hooking into API calls. It can steal HTTP secured connection session data. It can inject forms into legitimate web pages of banks by using webinjects. Such forms can include fields to entice the victim to enter data such as Personal Identification Numbers (PIN) that are not required not requested by the online financial institution. It can include keyloggers to capture legitimate data entered by the victim. It can include data mining algorithms to collect and forward only relevant, filtered data to the Command and Control server via encrypted data channels.

Crimeware Toolkit Components

An examination of crimeware toolkits conducted for the purpose of this research reveals that a number of high level components appear to be in common to most of the variants and versions. This commonality includes configuration files for customizing the botnet and other files, such as the webinjects file that contains content injection rules. There is typically a builder program that generates the malware binary to infect the victims from the clear text configuration files that are customised by the cybercriminal. The format of these configuration files vary between the variants of crimeware toolkits, but all of the toolkits examined used configuration files. An encrypted version of the configuration file is created using an encryption key. It is a separate file to the executable and is generally downloaded during execution of the binary. The behaviour of the binary on the target system can then be modified at the direction of the cybercriminal. A small number of standalone programs are also included in the toolkits, including file archivers, build tools, packers, protectors, assemblers and a PHP compiler for compiling PHP web scripts. Other tools for deploying the malware may also be found as well as supporting documentation such as manuals to assist in the authoring process.

The very nature of the development and release process is tailored to ensure that it works for the cybercriminal who is using the crimeware toolkit to create the banking trojan with minimal effort and complications. This means that there is a definitive structure to the configuration files for the build process and that various artefacts of the development and release process can be recovered to reconstruct the event of having built the resultant banking trojan. Various plug-ins, enhancements and customisations are available and can be purchased and traded on underground forums (Hypponen, 2011), but it could be expected that the core framework, and development and release process, for any of the particular versions or variants of the crimeware toolkits will remain consistent in the short term. This is essentially because a level of customisation is essential for customers to tailor the trojan for their particular needs. This necessitates an editable configuration file. This in turn means the configuration file requires structure with defined fields, so that the builder program can parse and interpret it to create a trojan that will function correctly. It would be very difficult to consider an alternate method that combines ease of use, consistency and reliability. In addition, reuse of tried, true and tested code is a fundamental principal of best practice software engineering.

FORENSIC ANALYSIS

Crimeware Toolkit Artefacts

Detection of the tools in the toolkit may not be reliably detected by an antivirus (AV) software suite. This is because AV software that is signature based is reliant upon previous detection and extraction of an appropriate signature. The tool can be protected and/or packed which will obfuscate the code, change the hash and change the signature of the code. A variety of techniques can be implemented to further hinder the digital forensic analyst. This can include techniques such as anti emulation, anti online analysis, anti hardware, anti debugger, anti disassembler, anti tools, anti memory, anti process and rootkits as discussed by Brand, Valli and Woodward (2010). In addition, it is quite simple to change the hash of any program to be the hash of a program that is on a known good file list to avoid being relegated to a list of unknown files for investigation (Foster, Liu, 2005).

The configuration files between the toolkits can be different, but they do appear very similar within variants and versions of the same toolkit with clearly defined key fields and parameters. Although these files may be deleted, the potential exists to recover full or remnant parts of the files from memory devices, allocated or unallocated space, the hibernation files, the memory page files and from physical memory dumps. The configuration files are typically textual in nature, and lend themselves to key word searches. The tailoring of the banking trojan itself is determined by the configuration files. Hence to determine the released trojans capability, recovery of the configuration file could provide supporting evidence of the activity of having built a particular banking trojan. In addition, it could be possible to associate a particular banking trojan with a particular configuration file. Figure 4 demonstrates the structure of the ZeuS 1.2.4.2 configuration file. Clearly defined fields and delimiters are present. Figure 5 presents a small section of the WebInjects file. The WebInjects clearly lists URLs of common and popular banking websites. Figure 6 shows a small subsection of the user manual that uses terms associated with the particular crimeware kit version. The manual needs to be descriptive enough for users to tailor their developed malware. In all cases of the selection of crimeware toolkits examined, keywords and structure are evident and could be very useful for keyword searches, file carving, indexing and filtering.

```
;Build time:   14:15:23 10.04.2009 GMT
;Version:      1.2.4.2

entry "StaticConfig"
  ;botnet "btn1"
  timer_config 60 1
  timer_logs 1 1
  timer_stats 20 1
  url_config "http://localhost/config.bin"
  url_compip "http://localhost/ip.php" 1024
  encryption_key "secret key"
  ;blacklist_languages 1049
end

entry "DynamicConfig"
  url_loader "http://localhost/bot.exe"
  url_server "http://localhost/gate.php"
  file_webinjects "webinjects.txt"
  entry "AdvancedConfigs"
    ;"http://advdomain/cfg1.bin"
  end
  entry "WebFilters"
    "!*.microsoft.com/*"
    "!http://*myspace.com*"
    "https://www.gruposantander.es/*"
    "!http://*odnoklassniki.ru/*"
    "!http://vkontakte.ru/*"
    "@*/login.osmp.ru/*"
    "@*/atl.osmp.ru/*"
  end
  entry "WebDataFilters"
    ;"http://mail.rambler.ru/*" "passw;login"
  end
  entry "WebFakes"
    ;"http://www.google.com" "http://www.yahoo.com" "GP" "" ""
  end
  entry "TANGrabber"
    "https://banking.*.de/cgi/ueberweisung.cgi/*" "S3R1C6G" "*&tid=*" "*&betrag=*"
    "https://internetbanking.gad.de/banking/*" "S3C6" "*" "*" "KktNrTanEnz"
    "https://www.citibank.de/*/jba/mp#/SubmitRecap.do" "S3C6R2" "SYNC_TOKEN=*" "*"
  end
  entry "DnsMap"
    ;127.0.0.1 microsoft.com
  end
end
```

Figure 4 ZeuS 1.2.4.2 configuration file highlighting keywords

```
set_url https://banking*.anz.com/* GPL
data_before
<td class="actionHeaderTopPadding" >Balances and Transactions
data_end
data_inject
data_end
data_after
<!--This is required for bway:button
data_end

set_url https://olb2.nationet.com/signon/signon* GPL
data_before
Passnumber:</SPAN>
data_end
data_inject
data_end
data_after
<TD COLSPAN="4" CLASS="consolebackground">
data_end
set_url https://www.nwolb.com/Login.asp* GPL
data_before
<span OnClick="window.open('https://www.nwolb.com/help.asp
data_end
data_inject
data_end
data_after
<script language="javascript">document.write("<img src='brands/NWB/
data_end
```

Figure 5 small extraction of a ZeuS 1.2.4.2 WebInjects file highlighting keywords

```
User's Guide (Draft)
************************************


================
= Contents =
================


1. Description and features.
2. Setting up the server.
   2.1. HTTP-server.
   2.2. The interpreter PHP.
   2.3. MySQL-server.
   2.4. Control Panel.
     2.4.1. Installation.
     2.4.2. Update.
     2.4.3. File / system / fsarc.php.
3. Setting Bot.
4. Working with BackConnect.
5. Changelog.
6. F.A.Q.
7. Myths.
```

Figure 6 Small extraction of the ZeuS 1.2.4.2 manual

Various artefacts from memory dumps may be extracted, such as the key entered via the keyboard for encrypting the configuration file, text entered via the keyboard, configuration files loaded into memory by the tools, and remnants of supporting tools loaded into memory.

Web browsers act as an interface to internet activities associated with the tailoring and deployment of malware, including the reading of user's manuals and potentially online support via various mechanisms. The browser and other internet artefacts may be extracted to assist in the reconstruction of events. This may include artefacts from chat programs, file sharing programs, web based mail clients, web browsing history, downloads, cookies, cache, form history, favourites and profiles. Web history can include redirects, visit from, visit from a bookmark, visit from a typed URL, form visits, hidden visits and thumb nailed pages. Download history could include plain text, images, media, PDFs, manual downloads, files downloaded to non-standard locations.

PRESENTATION OF EVIDENCE

Computer evidence needs to be admissible, authentic, complete, reliable and believable (The Internet Society, 2002). This paper has considered the acquisition phase of the investigation to be out of scope, however the resulting output of the analysis phase will become the input to the presentation phase and hence the analysis phase must be conducted with the requirements of admissibility in mind. Given that a set sequence of events, using various tools and configuration files must be employed to produce the banking trojan, and will produce artefacts in a variety of locations and formats, the author proposes that presentation of evidence would be enhanced by presenting the evidence in a timeline that maps to the sequence of events required to build the trojan. Such an approach may provide focus in recovering evidence in the analysis phase of the investigation and could be used in future investigations to assist in the analysis phase. This approach may also assist to show *actus reus* and *mens rea.* A generic template for this activity is provided in Figure 7. Evidence associated with each event in the build process can be determined and documented in a timeline format.

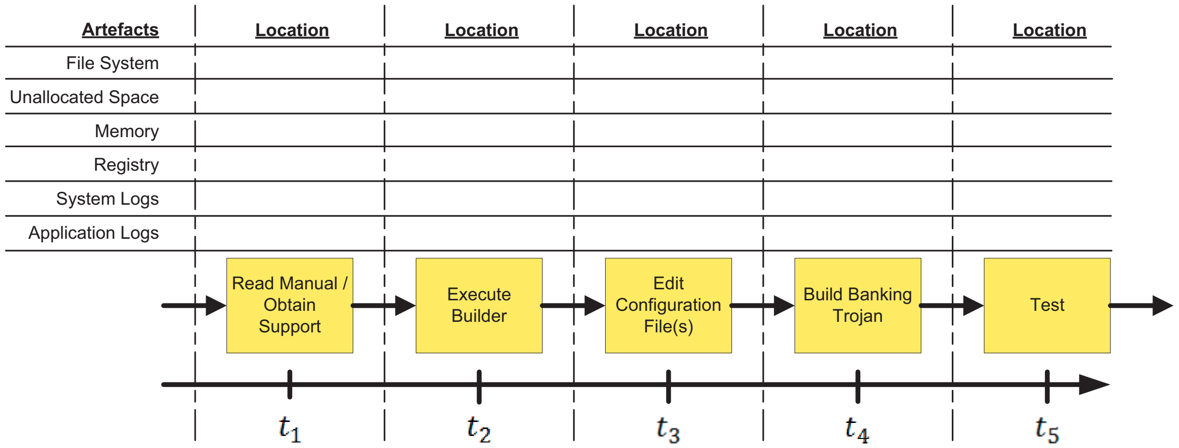

Figure 7 Timeline and Evidence Location Correlation

CONCLUSION

The population of the world is increasingly coming online, transferring traditional models of conducting business to the internet, necessitating the transfer and storage of personally identifiable information and financial data. This provides cybercriminals with the motivation to transfer traditional models of conducting fraud to the internet to take advantage of such transactions. Crimeware toolkits, through a simple graphical user interface to produce a banking trojan, provide an opportunity to harvest financial credentials such that funds can be withdrawn from the accounts of their victims. This research has shown that although there are many versions and variants of crimeware toolkits available, the underlying process of building the trojan using various tools and configuration files follows a generic process that leaves artefacts in various locations. To this end, future work will lead to developing sequence diagrams for building the versions and variants of known trojans using the required tools and configuration files that map to the sequence of events associated with the building process. It is intended that a database of information be constructed using this research that can assist the forensic investigator in the analysis phase of an investigation that hypothesises that a crimeware tool kit has been employed to author a banking trojan on a computer system.

REFERENCES

Barrett, J., (2011). Spy Eye and Carberp - the new banker trojans offensive. Retrieved October 16, 2011 from http://cleanbytes.net/spy-eye-and-carberp-the-new-banker-trojans-offensive

Brand, M. (2010). Analysis Avoidance Techniques of Malicious Software, Edith Cowan University, Perth, WA.

Brand, M., Valli, C., Woodward, A., (2010). Lessons Learned from an Investigation into the Analysis Avoidance Techniques of Malicious Software. Retrieved October 16, 2011 from http://ro.ecu.edu.au/adf/74/

Foster, C., Liu., V., (2005). Catch me, if you can. Retrieved October 16, 2011 from http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-foster-liu-update.pdf

Hypponen, M., (2011). Fight Cybercrime, But Keep the Net Free. Retrieved October 16, 2011 from http://www.f-secure.com/weblog/archives/00002210.html

Norton (2011). Cyber Crime Report 2011. Retrieved October 16, 2011 from http://www.symantec.com/content/en/us/home_homeoffice/html/ncr/

S21sec (2010). ZeuS Mitmo: Man-in-the-mobile. Retrieved October 16, 2011 from http://securityblog.s21sec.com/2010/09/zeus-mitmo-man-in-mobile-i.html

Shinder, D., (2002). Scene of the Cybercrime - Computer Forensics Handbook. Syngress.

The Internet Society, (2002). RFC 3227 Guidelines for Evidence Collection and Archiving. Retrieved November 17, 2011 from http://www.ietf.org/rfc/rfc3227.txt

Trend Micro (2011a). 1Q 2011 Crimeware Report. Retrieved October 16, 2011 from http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/1q_2011_crimeware_report.pdf

Trend Micro (2011b). 2Q 2011 Crimeware Report. Retrieved October 16, 2011 from http://us.trendmicro.com/imperia/md/content/us/trendwatch/researchandanalysis/2q_2011_crimeware_report.pdf

Trend Micro (2011c). Soldier Spy Eyes a Jackpot. Retrieved October 16, 2011 from http://blog.trendmicro.com/soldier-Spy Eyes-a-jackpot/