Digital camouflage: crypters make malware undetectable

Australian Government

Australian Signals Directorate

ASD AUSTRALIAN SIGNALS DIRECTORATE

ACSC Australian Cyber Security Centre

AFP

Google Cloud

NCA

National Crime Agency

NEW ZEALAND POLICE

Ngā Pirihimana o Aotearoa

Table of contents

Context

Cybercriminals use a range of **malware** in a variety of ways that can cause significant cyber security incidents. To counter this, the cyber security industry shares information about malware so that defenders can create rules to detect this malicious software in their own systems. Security software and platforms such as VirusTotal – a service that analyses files and URLs for malicious content – identify and share detection information faster than ever before. This sharing helps strengthen system defences, making it harder for cybercriminals to operate effectively.

Because of this, some cybercriminals are turning to '**crypters**'. Crypters hide malware, which make it more difficult for security software to detect. This advisory raises awareness of crypter activity and provides readers with an understanding of crypter services, how they operate and the risks they pose.

Introduction

As base platform protections and malware detection services become more advanced, cybercriminals seek out new ways to deliver and run malware on target systems without detection. **Crypters** help them do this by transforming malware files so that they can run on a victim's system whilst remaining "fully undetected", or "FUD".

Crypters are financially motivated cybercriminals who typically advertise their services on dark web forums. Cybercriminals who have obtained malware for distribution may also purchase **crypter services** to improve the chances of success of their campaigns.

Crypters hide malware by processing the malicious files through specialised **crypter software**. Crypter software scrambles the contents of a file using a variety of obfuscation, anti-analysis, anti-detection and cryptographic techniques. This process creates a new '**crypted malware**' file that hides its content and behaviour from detection services more effectively. All malware has distinct content and behaviour, which creates a detection 'fingerprint'; however, malware that has been "crypted" is designed to avoid creating a detectable fingerprint.

Crypter activity poses risks at two levels. **For organisations**, malware that bypasses system defences may remain active for longer periods and lead to a cybersecurity incident, such as unauthorised access to sensitive information, financial loss or operational downtime. The risks will change depending on the type of malware and goals of the cybercriminal. **For the broader cyber defensive community**, crypter activity aims to undermine the effectiveness of detection sharing. By the time a crypted malware file is discovered, and a detection is shared, the malware may have already been '**re-crypted**' (processed through crypter software again). This means a detection may prove effective for only a short period of time against a specific file, driving up the volume and rate of detection sharing while reducing the operational value of each detection. Crypter services present a constant race between detection services and crypters adapting their crypter software.

This advisory provides an overview of how crypters monetise their services, how crypter software works, how crypters and cybercriminals check if a malware file is detectable and how to impact crypter services, including mitigations advice for organisations and cyber security professionals.

Crypter-as-a-service

Malware detection services and open-source cyber threat intelligence platforms such as VirusTotal, extract and share the details and indicators of compromise (IOCs) associated with malware faster than ever before. This poses a challenge for cybercriminals.

Crypter-as-a-service is a response to the increased sophistication of malware detection, and the operational requirements of cybercriminals that rely on successful malware deployments to generate financial gain.

The crypter-as-a-service model targets cybercriminals who do not have the skills to obfuscate malware, which includes both experienced, and less capable cybercriminals with minimal technical knowledge. Crypter services lower the barrier to entry to participating in cybercriminal activity as it provides an easy and often automated service that assists in improving the success rate of malware deployment. Most crypters offer a customer service channel and often declare that once a file produced by their crypter is detected by AVs, they will **crypt** the file again (apply cryptography techniques to make it 'fully-undetectable' again).

Crypted malware is predominantly used to:

- **Evade detection**: AV software is less likely to detect crypted malware, allowing it to execute on systems undetected.
- **Prolong malware lifespan**: By frequently changing the encryption or obfuscating the files, attackers can extend the lifespan of their malware, making it harder for security researchers to develop effective countermeasures.
- **Increase profitability**: Successful evasion of detection increases the likelihood of malware deployment, leading to higher profitability for cybercriminals.

Crypters typically advertise, sell and distribute crypter services through various dark web marketplaces, channels and forums. Crypters may automate their services via intuitive Telegram bots and websites that facilitate file transfers. Some crypters prefer to sell directly to trusted clients, ensuring a higher level of security and exclusivity.

Figure 1. Example of crypter services advertisement on the dark web

The pricing model of crypter services can vary depending on a variety of factors, including subscription time frame (including weekly, monthly, annually, and per crypt), cryptographic services offered, and the presence of premium offers, including 24/7 support and autocrypt (automated crypt services). More complicated obfuscation techniques and rapid services command prices in the thousands, while single automated crypts start at US$25. The business models for distributing crypter services can vary, but commonly include:

- **Subscription services**: Cybercriminals offer ongoing access to crypter services for a recurring fee, ensuring a steady stream of income.
- **One-time purchases**: Single file crypt for a one-time fee, appealing to cybercriminals looking for a quick solution.
- **Custom solutions**: High-end crypter services may offer custom solutions tailored to the specific needs of their clients, commanding higher prices.

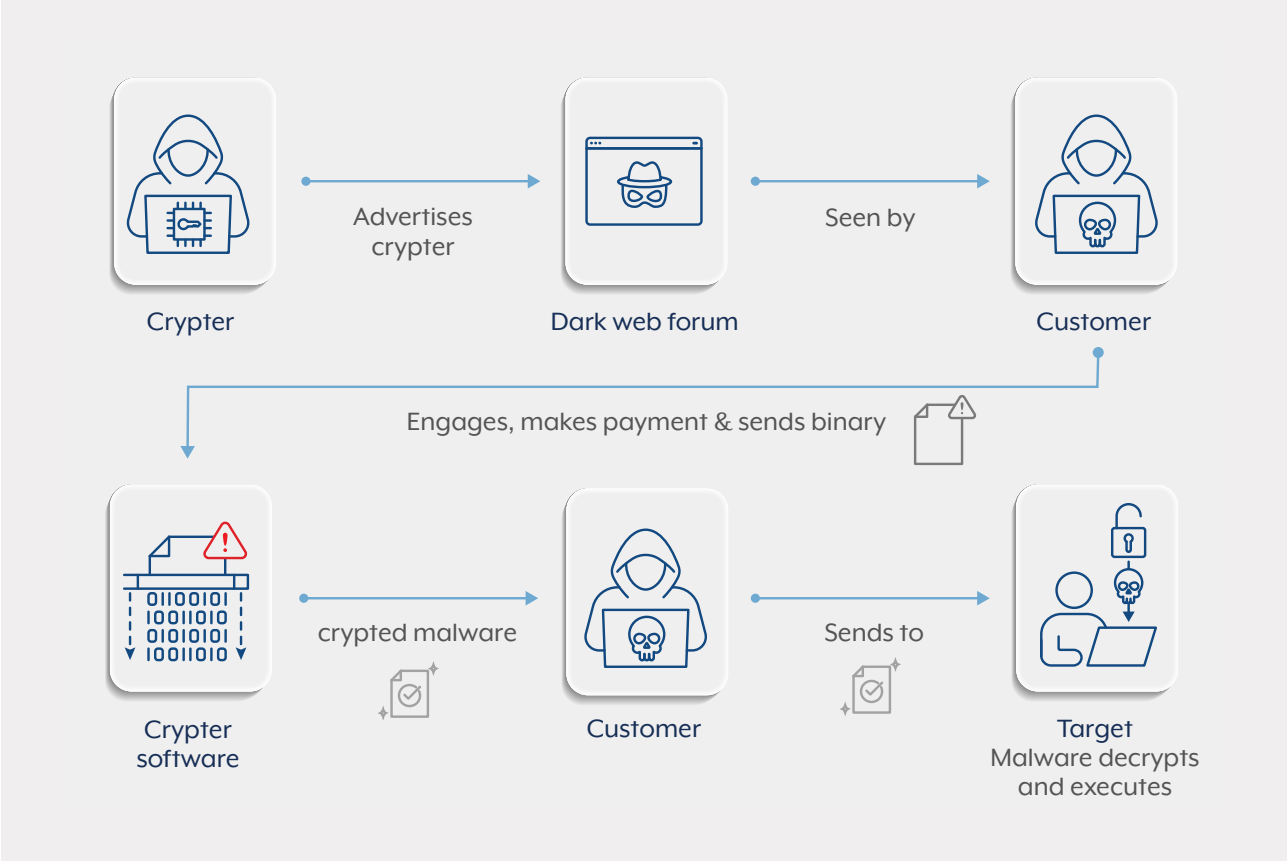

Figure 2. Crypter-as-a-service life cycle

Crypter software

Crypter services play a crucial role in helping malware evade detection at various stages, including pre-execution (static analysis), runtime, and post-execution (behavioural analysis). Static analysis inspects a file for known malicious signatures or patterns, without executing it. Behavioural analysis monitors how a program behaves when executed on a system, to identify malicious activities. Crypters aim to prevent malware from detection during both of these stages. Some crypters include features that identify if the malware is being executed in a virtual machine (VM) or being debugged and, if these conditions are detected, the malware may alter its behaviour or terminate to avoid analysis.

Crypter software is commonly composed of two parts, the '**builder**' and the '**stub**' which work together to obfuscate, encrypt and deliver malware.

The **builder** is the main interface or tool that allows cybercriminals to configure and customise their crypt. It provides options for encryption and obfuscation settings and then compiles the malware with the selected settings to generate a new version of the malware, the 'crypted malware'.

The **stub** is a small piece of unencrypted code attached to the encrypted malware. The stub runs first and exists to decrypt and execute the original malware payload when the crypted file runs on a target system.

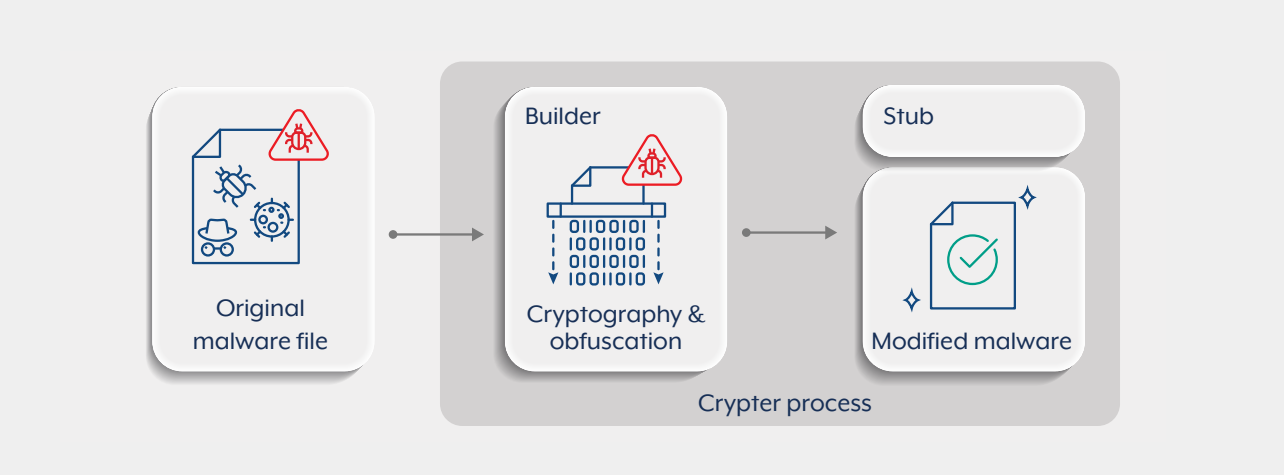

Figure 3. Summary of the crypter process

Obfuscation and cryptography techniques offered by crypters vary and may include:

- **Code obfuscation**: Also known as binary obfuscation, this alters the code of the malware to make it less recognisable to security software. Binary obfuscation is widely used in legitimate applications to protect intellectual property (eg. gaming and creative software). Crypters typically combine binary obfuscation with other methods to hide malicious processes.
- **Metamorphic obfuscation**: Rewrites the malware code with each iteration, making a new version of the malware.
- **Polymorphic encryption**: Changes the encryption key and creates a unique version with each iteration, making it difficult for AV software to detect and track malware campaigns.

- **Use of packers**: Packers compress and encrypt executable files to reduce their size and obfuscate their contents. Legitimate use of packers is common, such as reducing the size of software distributions. Crypter software provides a packer that modifies and hides malware files to evade AV detection. Packers offered by crypters range in uniqueness from standard static packers, or customisable packers, to bespoke packers created specifically for individual customers on a file-by-file basis.

An encryption key is a critical piece of data used within an encryption algorithm, to convert plaintext into unreadable ciphertext, and back again. The encryption key ensures the contents remain private and accessible to key holders only. Keys come in asymmetric pairs (public and private) or as a symmetric single key.

Developing custom crypter software requires a high level of technical skill often reflected by higher prices. Some of the capabilities offered by crypter services include:

- support for multiple CPU architectures, including 32, 64 bit and x86 architectures
- support for .NET framework
- file binding, file padding/junk code
- process hollowing
- malicious code injection into legitimate windows processes
- dynamic link library (DLL) side loading
- user account control (UAC) bypass
- XLL exploits (Excel specific DLL's)
- anti-virtual machine and sandbox evasion
- delay on start up
- dynamic URL payload delivery techniques
- changing file icon and file size.

For additional information about the above capabilities please refer to **Annex A** at the end of this advisory.

Crypted malware and counter-antivirus (CAV) services

Crypters may take an additional step to ensure their software produces what the cybercriminal considers to be "fully undetectable" malware. After generating crypted malware, they verify its detectability using cybercriminal-operated antivirus services, known as counter-antivirus (CAV) services. CAV services offer to scan malware files to check if legitimate antivirus solutions detect it, and then returns the results to the customer. CAV services are desirable to cybercriminals as they will not share the details of any detected malware with global cyber threat intelligence feeds. Cybercriminals often use CAV and crypter services concurrently to gain confidence that malware will successfully reach target systems.

Cybercriminals may approach crypter services with distrust if they are unable to assess the detectability of the crypted file on their own. To assure clients of their effectiveness, crypter services often include the CAV scan results with the crypted file as evidence of successful antivirus evasion.

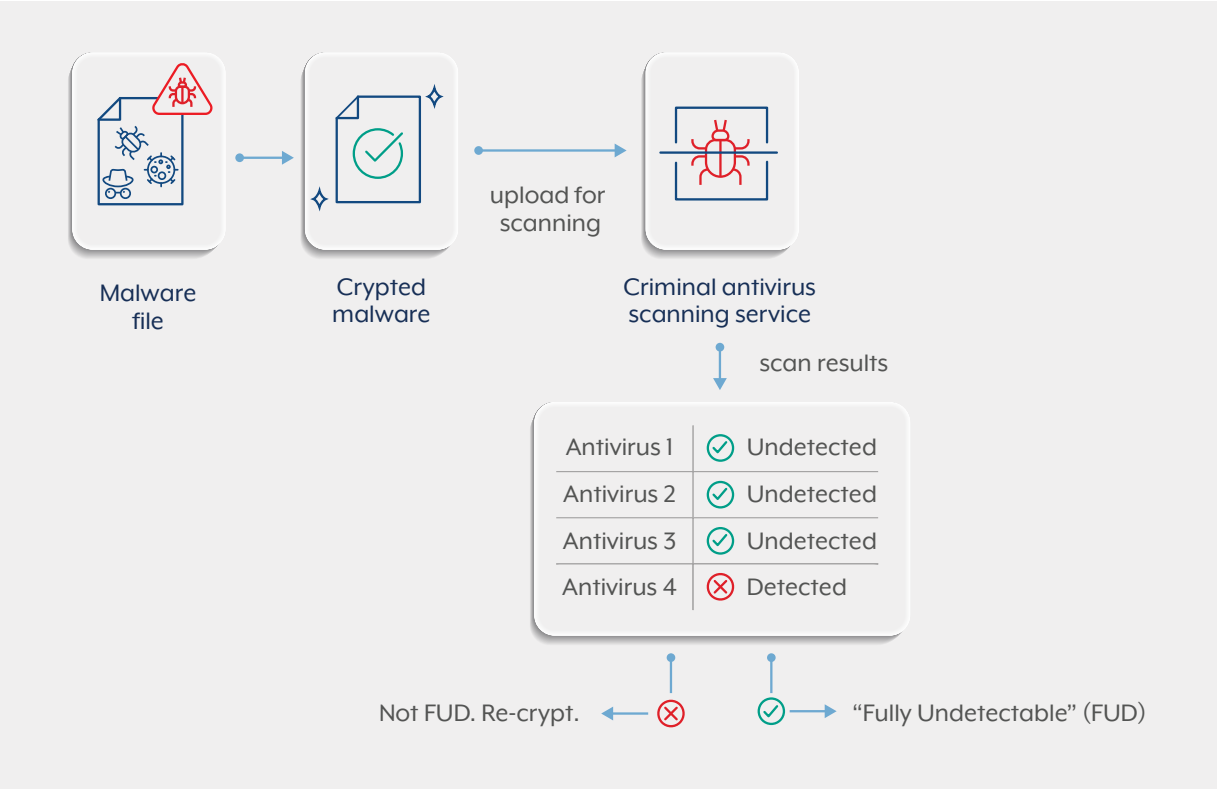

Figure 4. Malware is crypted before uploading to a CAV service for scanning. The results of the CAV scan are indicative of detection by legitimate defensive software.

Fictional case study

On the dark web, Jane, a seasoned cybercriminal, discovered an ad for a new crypter service claiming to make malware "fully undetectable". Jane decides to test the crypter service against a target organisation that she has been trying to gain initial access into without success.

Jane prepares an information stealer payload to steal credentials from the organisation's system. Using the crypter service, Jane engages with a crypter bot to specify the customisations required, and then uploads the information stealer file to be crypted.

Jane receives a new crypted malware file back. The process altered the malware's signature, making it virtually unrecognisable to traditional detection methods.

To confirm the file evades detection, Jane utilises a CAV service. Jane trusts the CAV service with her file and knows that if the scan detects malicious contents, the CAV will not share the details with global cyber threat intelligence feeds. The CAV service scans the crypted malware and reports no detection, bolstering Jane's confidence.

Jane creates a convincing phishing email containing an attachment with the crypted malware, which she disguises as a legitimate communication from a trusted supplier. Jane sends the phishing email and several unsuspecting employees fall for the ruse, opening the attachment and unknowingly executing the malware.

The crypter service proved its worth, as the organisation's antivirus software failed to detect the intrusion. Jane successfully infiltrated the organisation, evading their security measures and exfiltrating valuable data.

Implications

One crypter service can crypt a single malware file many times to generate hundreds of unique file signatures. Furthermore, the automation of crypter services made available via Telegram bots and websites may facilitate the generation of a large volume of potentially undetectable malware files.

With a crypter-as-a-service model, cybercriminals are not required to customise or repeatedly purchase new malware files from a malware developer. As such, crypter services lower the barrier for entry-level cybercriminals to successfully deliver malware and cause widespread harm.

Crypted malware evades detection, meaning that seemingly harmless downloaded files may be performing malicious functions without the user's knowledge. Organisations should be mindful that legitimate antivirus scanning services may be unable to detect the malicious content of a file due to these crypter techniques.

A single successful malware deployment on corporate systems made possible by crypted malware bypassing system defenses may result in unauthorised access to corporate data, credential theft, financial losses, operational downtime, ransomware or other cyber security incidents.

How to impact crypter services

Crypter services present a constant race between detection mechanisms and crypters adapting their crypter software. Crypter services reach a global cybercriminal market and are often hosted in permissive jurisdictions. International co-operation is essential to address the problems posed by crypters. A sustained approach that degrades, disrupts and denies crypter services is required.

Reverse engineering efforts from malware analysts to identify file signatures are likely outpaced by the adaptability of crypter services. Efforts from malware analysts within the cyber security industry should focus on behavioural analysis when reverse engineering the crypter techniques applied to a file. Making undetectable malware gives the illusion of impossibility, but thorough analysis can reveal the techniques involved.

Mitigations

Application control

- Implement application control on workstations, internet facing servers and internal servers (ISM-0843, ISM-1490, ISM-1656).

Many security products rely on signatures to detect malicious code. This approach is only effective when malicious code has already been profiled, and signatures are available from security vendors. Unfortunately, malicious actors can easily create variants of known malicious code to bypass traditional signature-based detection.

Endpoint detection and response

- Deploy endpoint detection (EDR) or Host-based Intrusion Prevention System (HIPS) on all workstations (ISM-1341), critical servers and high value servers (ISM-1034).

A Host-based Intrusion Prevention System (HIPS) or Endpoint Detection and Response (EDR) solution can use behaviour-based detection to assist in identifying and blocking anomalous behaviour as well as detecting malicious code that security vendors are yet to identify. It is important that either a HIPS or EDR solution is implemented on workstations, critical servers and high-value servers.

Antivirus configuration

- Implement antivirus application on workstations and servers (ISM-1417) and configure it to:
  - enable signature-based detection and set to high sensitivity
  - enable heuristic detection (identifies suspicious patterns) and set to high sensitivity
  - enable reputation checking (flags files from untrusted sources)
  - enable ransomware protection
  - update signatures at least daily
  - schedule regular full-system scans of all drives and removable media.

System maintenance

- Keep operating systems up to date (ISM-1415, ISM-1420, ISM-1418, ISM-1419). Apply patches promptly to ensure the latest security protections are in place.

Advanced detection techniques

- Build behaviour-based detections that enable behavioural scanning: Build detections around common crypter techniques, such as checking for sandboxes and file padding/ junk code.

Modern malware often obfuscates itself until the moment it runs, making static file scanning ineffective. Shift priorities from static disk analysis to runtime memory scanning. Since runtime-crypted payloads decrypt their malicious contents entirely in memory to evade disk scanners, identifying malicious or suspicious memory allocations, reflective code loading, and process hollowing/manipulation is critical.

For more information

- Australian Signals Directorate's *Information security manual*
- New Zealand Information security manual

Annex A

Capability offered	Description
Supports multiple CPU architectures	The crypter is capable of creating a crypted executable that can run on various CPU architectures, which supports malware to execute successfully on a wider range of systems, increasing its spread and impact.
Supports .NET framework	The crypter is capable of creating a crypted executable that can execute within the .NET framework environment.
File binding	Combining multiple files into a single executable to evade detection and distribute malware. By binding malicious files with legitimate ones, attackers can trick users into executing malware as the combined file looks harmless.
Process hollowing	A technique where a new process is created in a suspended state, its memory replaced with malicious code, and then the process is resumed. This allows malware to hide within legitimate processes, evading detection by security software that only scans for malicious executables. Malware uses this method to hide its activity from system defences.
Malicious code injection into legitimate Windows processes	Injecting malicious code into legitimate processes to execute harmful actions without detection. Code injection enables malware to hide within legitimate processes, making it difficult for security tools to distinguish between benign and malicious activities.
Dynamic Link Library (DLL) side loading	Loading a malicious DLL instead of the intended legitimate one, to execute malicious code. DLL side loading exploits the trust relationship between applications and their dependencies, enabling malware to load and execute without raising suspicion.
User Account Control (UAC) bypass	Methods to bypass the User Account Control, which aims to prevent unauthorised changes to the system. Bypassing UAC can lead to unauthorised elevation of privileges without alerting the user, enabling it to make system-level changes and install additional malicious components.

Capability offered	Description
XLL exploits (Excel specific DLL's)	Exploiting vulnerabilities in Excel by using malicious XLL files. This method enables malware to spread through seemingly innocuous documents and evade detection by traditional security measures.
Anti-Virtual machine and sandbox evasion	Techniques to detect and avoid execution in virtual machines or sandbox environments means the malware can prevent its behaviour from being analysed by security researchers.
Delay on startup	Delaying the execution of malicious code may help the malware to evade detection during the initial startup phase. This potentially allows malware to bypass initial security scans and heuristic analysis, increasing the likelihood of successful infection.
Dynamic URL payload delivery techniques	Using dynamically generated URLs to deliver malicious payloads, making it harder for defenders to block or analyse them. Dynamic URLs make it difficult for security tools to blacklist or analyse them as the URLs change frequently.
Changing file icon and file size	Modifying the icon and size of malicious files to appear as legitimate files. Changing the appearance of malicious files makes them less suspicious to users and security tools.

Disclaimer

The material in this guide is of a general nature and should not be regarded as legal advice or relied on for assistance in any particular circumstance or emergency situation. In any important matter, you should seek appropriate independent professional advice in relation to your own circumstances.

The Commonwealth accepts no responsibility or liability for any damage, loss or expense incurred as a result of the reliance on information contained in this guide.

Copyright

Use of the Coat of Arms

The terms under which the Coat of Arms can be used are detailed on the Department of the Prime Minister and Cabinet website
https://www.pmc.gov.au/resources/commonwealth-coat-arms-information-and-guidelines

For more information, or to report a cyber security incident, contact us:

cyber.gov.au | 1300 CYBER1 (1300 292 371)

ASD AUSTRALIAN SIGNALS DIRECTORATE

ACSC Australian **Cyber Security** Centre