NETMANAGEIT

Intelligence Report

Kryptina RaaS | From Underground Commodity to Open Source Threat
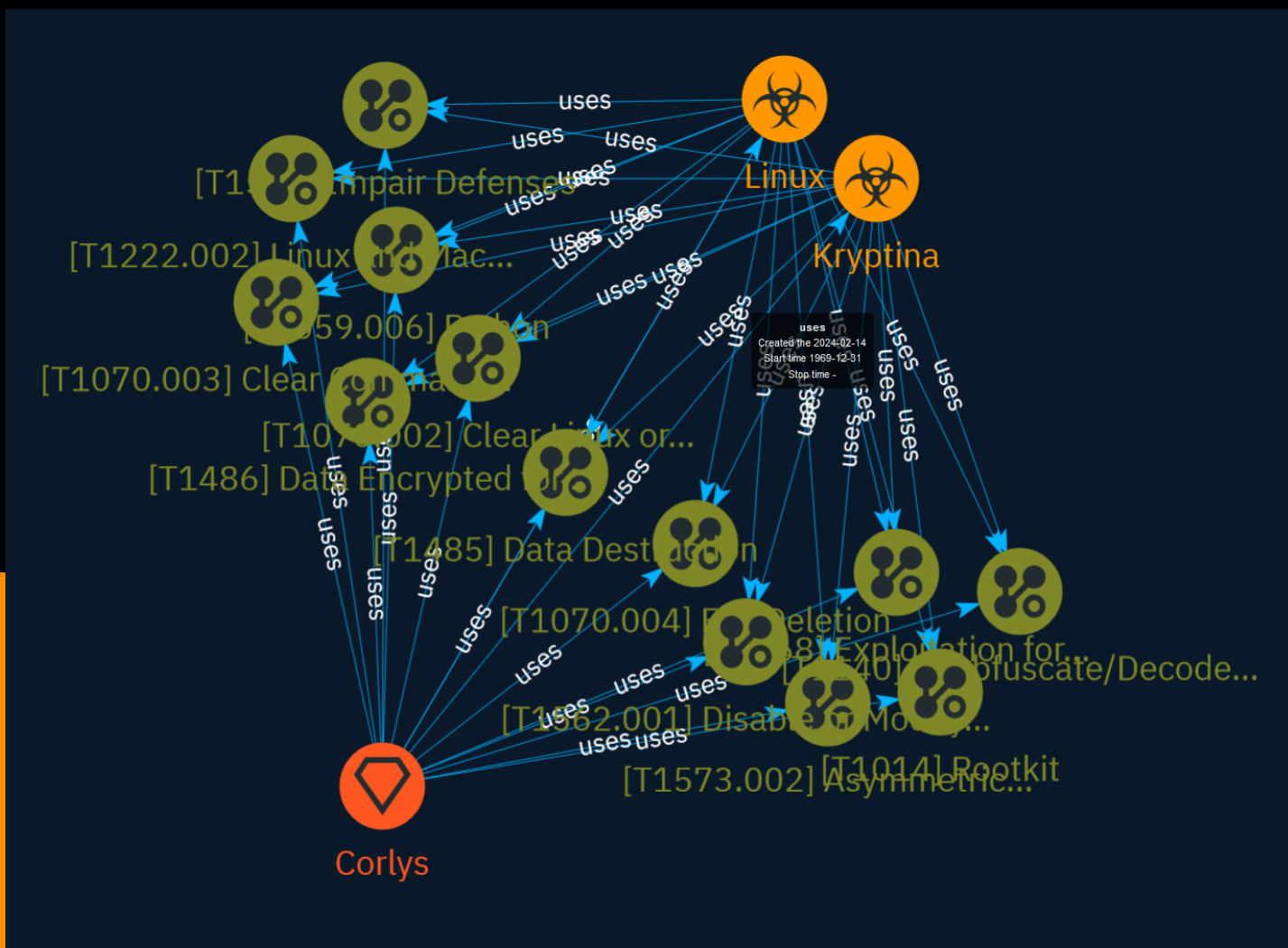
Table of contents

Overview

Entities

External References

Overview

Description

The recently observed Kryptina Raas, a dedicated Linux attack framework, has added a new twist to the ransomware-as-a-service model by moving from a paid service to an openly available tool. This increases the risk of proliferation of Kryptina variants and growth in attacks. Defenders need to protect Linux systems at the core of modern networks.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

15 / 100

Content

N/A

Malware

Name
Kryptina

Name
Linux

Intrusion-Set

Name
Corlys

Attack-Pattern

Name
File Deletion

ID
T1070.004

Description

Adversaries may delete files left behind by the actions of their intrusion activity. Malware, tools, or other non-native files dropped or created on a system by an adversary (ex: [Ingress Tool Transfer](https://attack.mitre.org/techniques/T1105)) may leave traces to indicate to what was done within a network and how. Removal of these files can occur during an intrusion, or as part of a post-intrusion process to minimize the adversary's footprint. There are tools available from the host operating system to perform cleanup, but adversaries may use other tools as well.(Citation: Microsoft SDelete July 2016) Examples of built-in [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059) functions include `del` on Windows and `rm` or `unlink` on Linux and macOS.

Name
Data Destruction

ID
T1485

Description

Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files or data on local and remote drives.(Citation: Symantec Shamoon 2012)(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017)(Citation: Unit 42 Shamoon3 2018)(Citation: Talos Olympic Destroyer 2018) Common operating system file deletion commands such as `del` and `rm` often only remove pointers to files without wiping the contents of the files themselves, making the files recoverable by proper forensic methodology. This behavior is distinct from [Disk Content Wipe](https://attack.mitre.org/techniques/T1561/001) and [Disk Structure Wipe](https://attack.mitre.org/techniques/T1561/002) because individual files are destroyed rather than sections of a storage disk or the disk's logical structure. Adversaries may attempt to overwrite files and directories with randomly generated data to make it irrecoverable.(Citation: Kaspersky StoneDrill 2017)(Citation: Unit 42 Shamoon3 2018) In some cases politically oriented image files have been used to overwrite data.(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017) To maximize impact on the target organization in operations where network-wide availability interruption is the goal, malware designed for destroying data may have worm-like features to propagate across a network by leveraging additional techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [OS Credential Dumping](https://attack.mitre.org/techniques/T1003), and [SMB/Windows Admin Shares](https://attack.mitre.org/techniques/T1021/002).(Citation: Symantec Shamoon 2012)(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017)(Citation: Talos Olympic Destroyer 2018). In cloud environments, adversaries may leverage access to delete cloud storage, cloud storage accounts, machine images, and other infrastructure crucial to operations to damage an organization or their customers.(Citation: Data Destruction - Threat Post)(Citation: DOJ - Cisco Insider)

Name

Python

ID

T1059.006

Description

Adversaries may abuse Python commands and scripts for execution. Python is a very popular scripting/programming language, with capabilities to perform many functions. Python can be executed interactively from the command-line (via the `python.exe` interpreter) or via scripts (.py) that can be written and distributed to different systems. Python code can also be compiled into binary executables. Python comes with many built-in packages to interact with the underlying system, such as file operations and device I/O. Adversaries can use these libraries to download and execute commands or other scripts as well as perform various malicious behaviors.

Name

Asymmetric Cryptography

ID

T1573.002

Description

Adversaries may employ a known asymmetric encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Asymmetric cryptography, also known as public key cryptography, uses a keypair per party: one public that can be freely distributed, and one private. Due to how the keys are generated, the sender encrypts data with the receiver's public key and the receiver decrypts the data with their private key. This ensures that only the intended recipient can read the encrypted data. Common public key encryption algorithms include RSA and ElGamal. For efficiency, many protocols (including SSL/TLS) use symmetric cryptography once a connection is established, but use asymmetric cryptography to establish or transmit a key. As such, these protocols are classified as [Asymmetric Cryptography](https://attack.mitre.org/techniques/T1573/002).

Name

Data Encrypted for Impact

ID

T1486

Description

Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done in order to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted.(Citation: US-CERT Ransomware 2016)(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017)(Citation: US-CERT SamSam 2018) In the case of ransomware, it is typical that common user files like Office documents, PDFs, images, videos, audio, text, and source code files will be encrypted (and often renamed and/or tagged with specific file markers). Adversaries may need to first employ other behaviors, such as [File and Directory Permissions Modification](https://attack.mitre.org/techniques/T1222) or [System Shutdown/Reboot](https://attack.mitre.org/techniques/T1529), in order to unlock and/or gain access to manipulate these files.(Citation: CarbonBlack Conti July 2020) In some cases, adversaries may encrypt critical system files, disk partitions, and the MBR.(Citation: US-CERT NotPetya 2017) To maximize impact on the target organization, malware designed for encrypting data may have worm-like features to propagate across a network by leveraging other attack techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [OS Credential Dumping](https://attack.mitre.org/techniques/T1003), and [SMB/Windows Admin Shares](https://attack.mitre.org/techniques/T1021/002).(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017) Encryption malware may also leverage [Internal Defacement](https://attack.mitre.org/techniques/T1491/001), such as changing victim wallpapers, or otherwise intimidate victims by sending ransom notes or other messages to connected printers (known as "print bombing").(Citation: NHS Digital Egregor Nov 2020) In cloud environments, storage objects within compromised accounts may also be encrypted. (Citation: Rhino S3 Ransomware Part 1)

Name

Exploitation for Privilege Escalation

ID

T1068

Description

Adversaries may exploit software vulnerabilities in an attempt to elevate privileges. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Security constructs such as permission levels will often hinder access to information and use of certain techniques, so adversaries will likely need to perform privilege escalation to include use of software exploitation to circumvent those restrictions. When initially gaining access to a system, an adversary may be operating within a lower privileged process which will prevent them from accessing certain resources on the system. Vulnerabilities may exist, usually in operating system components and software commonly running at higher permissions, that can be exploited to gain higher levels of access on the system. This could enable someone to move from unprivileged or user level permissions to SYSTEM or root permissions depending on the component that is vulnerable. This could also enable an adversary to move from a virtualized environment, such as within a virtual machine or container, onto the underlying host. This may be a necessary step for an adversary compromising an endpoint system that has been properly configured and limits other privilege escalation methods. Adversaries may bring a signed vulnerable driver onto a compromised machine so that they can exploit the vulnerability to execute code in kernel mode. This process is sometimes referred to as Bring Your Own Vulnerable Driver (BYOVD).(Citation: ESET InvisiMole June 2020)(Citation: Unit42 AcidBox June 2020) Adversaries may include the vulnerable driver with files delivered during Initial Access or download it to a compromised system via [Ingress Tool Transfer](https://attack.mitre.org/techniques/T1105) or [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570).

Name

Disable or Modify Tools

ID

T1562.001

Description

Adversaries may modify and/or disable security tools to avoid possible detection of their malware/tools and activities. This may take many forms, such as killing security software processes or services, modifying / deleting Registry keys or configuration files so that tools do not operate properly, or other methods to interfere with security tools scanning or reporting information. Adversaries may also disable updates to prevent the latest security patches from reaching tools on victim systems.(Citation: SCADAfence_ransomware)

Attack-Pattern

Adversaries may also tamper with artifacts deployed and utilized by security tools. Security tools may make dynamic changes to system components in order to maintain visibility into specific events. For example, security products may load their own modules and/or modify those loaded by processes to facilitate data collection. Similar to [Indicator Blocking](https://attack.mitre.org/techniques/T1562/006), adversaries may unhook or otherwise modify these features added by tools (especially those that exist in userland or are otherwise potentially accessible to adversaries) to avoid detection.(Citation: OutFlank System Calls)(Citation: MDSec System Calls) Adversaries may also focus on specific applications such as Sysmon. For example, the "Start" and "Enable" values in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-Microsoft-Windows-Sysmon-Operational` may be modified to tamper with and potentially disable Sysmon logging.(Citation: disable_win_evt_logging) On network devices, adversaries may attempt to skip digital signature verification checks by altering startup configuration files and effectively disabling firmware verification that typically occurs at boot.(Citation: Fortinet Zero-Day and Custom Malware Used by Suspected Chinese Actor in Espionage Operation)(Citation: Analysis of FG-IR-22-369) In cloud environments, tools disabled by adversaries may include cloud monitoring agents that report back to services such as AWS CloudWatch or Google Cloud Monitor. Furthermore, although defensive tools may have anti-tampering mechanisms, adversaries may abuse tools such as legitimate rootkit removal kits to impair and/or disable these tools.(Citation: chasing_avaddon_ransomware)(Citation: dharma_ransomware)(Citation: demystifying_ryuk)(Citation: doppelpaymer_crowdstrike) For example, adversaries have used tools such as GMER to find and shut down hidden processes and antivirus software on infected systems.(Citation: demystifying_ryuk) Additionally, adversaries may exploit legitimate drivers from anti-virus software to gain access to kernel space (i.e. [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068)), which may lead to bypassing anti-tampering features.(Citation: avoslocker_ransomware)

Name

Impair Defenses

ID

T1562

Description

Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders

can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators. Adversaries may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out of a computer or stopping it from being shut down. These restrictions can further enable malicious operations as well as the continued propagation of incidents.(Citation: Emotet shutdown) Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

Name

Deobfuscate/Decode Files or Information

ID

T1140

Description

Adversaries may use [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is the use of [certutil](https://attack.mitre.org/software/S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file.(Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b` command to reassemble binary fragments into a malicious payload.(Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016)

Name

Clear Command History

ID

T1070.003

Description

In addition to clearing system logs, an adversary may clear the command history of a compromised account to conceal the actions undertaken during an intrusion. Various command interpreters keep track of the commands users type in their terminal so that users can retrace what they've done. On Linux and macOS, these command histories can be accessed in a few different ways. While logged in, this command history is tracked in a file pointed to by the environment variable `HISTFILE`. When a user logs off a system, this information is flushed to a file in the user's home directory called `~/.bash_history`. The benefit of this is that it allows users to go back to commands they've used before in different sessions. Adversaries may delete their commands from these logs by manually clearing the history (`history -c`) or deleting the bash history file `rm ~/.bash_history`. Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to clear command history data (`clear logging` and/or `clear history`).(Citation: US-CERT-TA18-106A) On Windows hosts, PowerShell has two different command history providers: the built-in history and the command history managed by the `PSReadLine` module. The built-in history only tracks the commands used in the current session. This command history is not available to other sessions and is deleted when the session ends. The `PSReadLine` command history tracks the commands used in all PowerShell sessions and writes them to a file (`$env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt` by default). This history file is available to all sessions and contains all past history since the file is not deleted when the session ends.(Citation: Microsoft PowerShell Command History) Adversaries may run the PowerShell command `Clear-History` to flush the entire command history from a current PowerShell session. This, however, will not delete/flush the `ConsoleHost_history.txt` file. Adversaries may also delete the `ConsoleHost_history.txt` file or edit its contents to hide PowerShell commands they have run.(Citation: Sophos PowerShell command audit)(Citation: Sophos PowerShell Command History Forensics)

Name

Clear Linux or Mac System Logs

ID

T1070.002

Description

Adversaries may clear system logs to hide evidence of an intrusion. macOS and Linux both keep track of system or user-initiated actions via system logs. The majority of native system logging is stored under the `/var/log/` directory. Subfolders in this directory categorize logs by their related functions, such as:(Citation: Linux Logs) * `/var/log/messages:`: General and system-related messages * `/var/log/secure` or `/var/log/auth.log`: Authentication logs * `/var/log/utmp` or `/var/log/wtmp`: Login records * `/var/log/kern.log`: Kernel logs * `/var/log/cron.log`: Crond logs * `/var/log/maillog`: Mail server logs * `/var/log/httpd/`: Web server access and error logs

Name

Rootkit

ID

T1014

Description

Adversaries may use rootkits to hide the presence of programs, files, network connections, services, drivers, and other system components. Rootkits are programs that hide the existence of malware by intercepting/hooking and modifying operating system API calls that supply system information. (Citation: Symantec Windows Rootkits) Rootkits or rootkit enabling functionality may reside at the user or kernel level in the operating system or lower, to include a hypervisor, Master Boot Record, or [System Firmware](https://attack.mitre.org/techniques/T1542/001). (Citation: Wikipedia Rootkit) Rootkits have been seen for Windows, Linux, and Mac OS X systems. (Citation: CrowdStrike Linux Rootkit) (Citation: BlackHat Mac OSX Rootkit)

Name

Linux and Mac File and Directory Permissions Modification

ID

T1222.002

Description

Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files.(Citation: Hybrid Analysis Icacls1 June 2018)(Citation: Hybrid Analysis Icacls2 May 2018) File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.). Most Linux and Linux-based platforms provide a standard set of permission groups (user, group, and other) and a standard set of permissions (read, write, and execute) that are applied to each group. While nuances of each platform's permissions implementation may vary, most of the platforms provide two primary commands used to manipulate file and directory ACLs: `chown` (short for change owner), and `chmod` (short for change mode). Adversarial may use these commands to make themselves the owner of files and directories or change the mode if current permissions allow it. They could subsequently lock others out of the file. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via [Unix Shell Configuration Modification](https://attack.mitre.org/techniques/T1546/004) or tainting/hijacking other instrumental binary/configuration files via [Hijack Execution Flow](https://attack.mitre.org/techniques/T1574).(Citation: 20 macOS Common Tools and Techniques)

Attack-Pattern

External References

- https://otx.alienvault.com/pulse/65ccf886d4642fa3ca857798