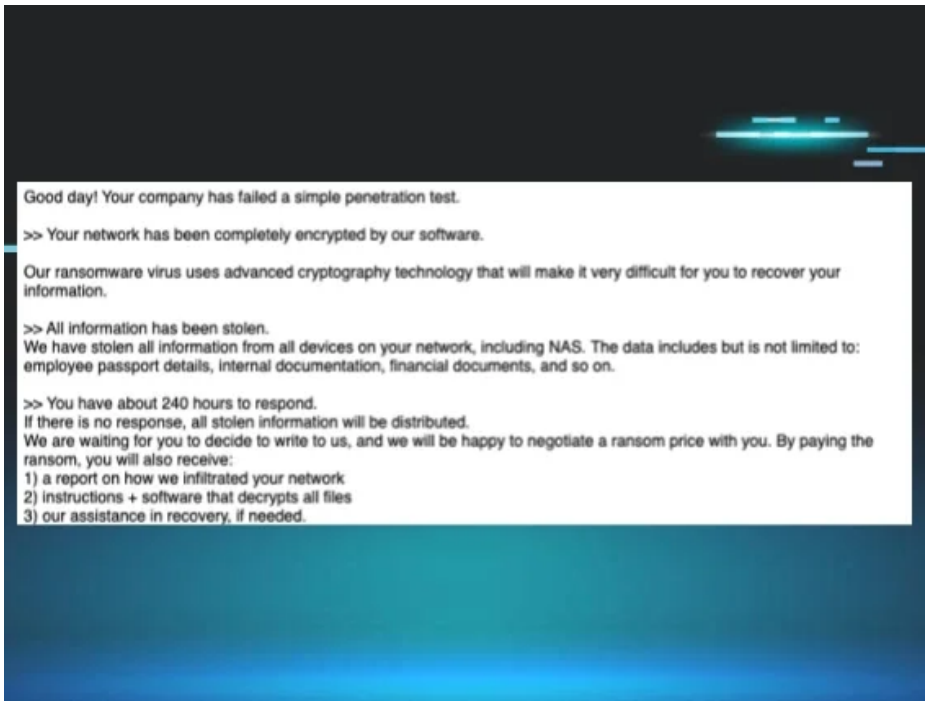
Obscura, an Obscure New Ransomware Variant

Harlan Carvey, Lindsey O'Donnell-Welch, Alden Schmidt, Anna Pham ⋮



On 29 August 2025, Huntress analysts encountered a previously unseen ransomware variant called “Obscura.” This name was taken from the ransom note (README_Obscura.txt), which also made several references to Obscura in its contents.

While researching this ransomware variant, analysts did not find any public references to a ransomware variant named Obscura.

The ransomware executable was first seen being executed across multiple hosts on the victim organization. This network had a limited deployment of the Huntress agent, which impacted both detection and response, inhibiting the SOC’s ability to respond effectively. This also limited our visibility into certain aspects of the attack, including the initial access vector.

However, what we were able to see was that the ransomware executable was found on the domain controller, in the path:

C:\WINDOWS\sysvol\sysvol\[domain].local\scripts\

In the incident observed by the Huntress SOC, the ransomware executable file was named for the domain in which it was found, in an apparent attempt to blend in (for this reason, we are not publicly identifying the name of this executable). The executable is a Go binary (including a Go build ID), and contains a number of file paths, such as:

/run/media/veracrypt1/Backups/Obscura/Locker/windows/locker/

/run/media/veracrypt1/Locker Deps/go1.15.linux-amd64/go/src/os/exec

The location of the binary on the domain controller was shared as the NETLOGON folder, which makes scripts and group policy objects (GPOs) available to users. In addition, the folder contents are automatically replicated across all domain controllers to maintain consistency. However, this also meant that the ransomware executable was automatically deployed throughout the infrastructure.

A scheduled task named SystemUpdate was created on multiple hosts throughout the network, including the domain controller, to execute the ransomware binary from the NETLOGON share.

On one of the user's machines, the threat actor created a scheduled task named "iJHcEkAG". The task runs the command cmd.exe /C netsh firewall set service type = remotedesktop mode = enable > \Windows\Temp\SJYfXB 2>&1 to enable Remote Desktop Protocol access through the Windows firewall.

When launched, the ransomware executable runs the following embedded command in an attempt to disable recovery on the endpoint:

cmd.exe /c vssadmin delete shadows /all /quiet

The ransom note itself is contained in the ransomware binary as a base64-encoded string.

Ransomware note contents

Good day! Your company has failed a simple penetration test.

>> Your network has been completely encrypted by our software.

Our ransomware virus uses advanced cryptography technology that will make it very difficult for you to recover your information.

>> All information has been stolen.

We have stolen all information from all devices on your network, including NAS. The data includes but is not limited to: employee passport details, internal documentation, financial documents, and so on.

>> You have about 240 hours to respond.

If there is no response, all stolen information will be distributed.

We are waiting for you to decide to write to us, and we will be happy to negotiate a ransom price with you. By paying the ransom, you will also receive:

1) a report on how we infiltrated your network

2) instructions + software that decrypts all files

3) our assistance in recovery, if needed.

>> They will not help you; they are your enemies.

Recovery agencies, the police, and other services will NOT HELP you. Agencies want your money, but they do not know how to negotiate.

If you think you can restore your infrastructure from external backups that we did not access, we warn you:

1) The laws of any country impose huge fines on companies for information leaks.
2) Playing against us will not work in your favor. We will gladly wipe every one of your servers and computers.

When you write to us, we expect to hear from you who you are and what your relationship to the company is.
Your ID: [REDACTED]
TOX: [REDACTED]
Blog: http://obscurad3aphckihv7wptdxvdnl5emma6t3vikcf3c5oiiqndq6y6xad.onion/
Obscura. 2025.

view raw README_Obscura.txt hosted with ❤ by GitHub

Technical analysis

When the binary is launched, it will check the status of an environment variable called DAEMON. If the value is 1, the binary will drop the ransom note and continue with encryption. If it's not present or has the value 0, it will run a series of functions to prepare the box for encryption. The main_run() function executes in daemon mode with DAEMON=1 set. It retrieves the threat actor's 32-byte public key by decoding a hardcoded base64 string embedded within the executable, then performs system reconnaissance by enumerating all storage devices and calculating their capacities to create a comprehensive map of all available drives and their storage sizes for encryption.

- The ransomware decodes a base64 encoded ransom note from the embedded data and writes it to C:\README-OBSCURA.txt. If decoding fails, it prints "failed to decode note: %s" and exits.

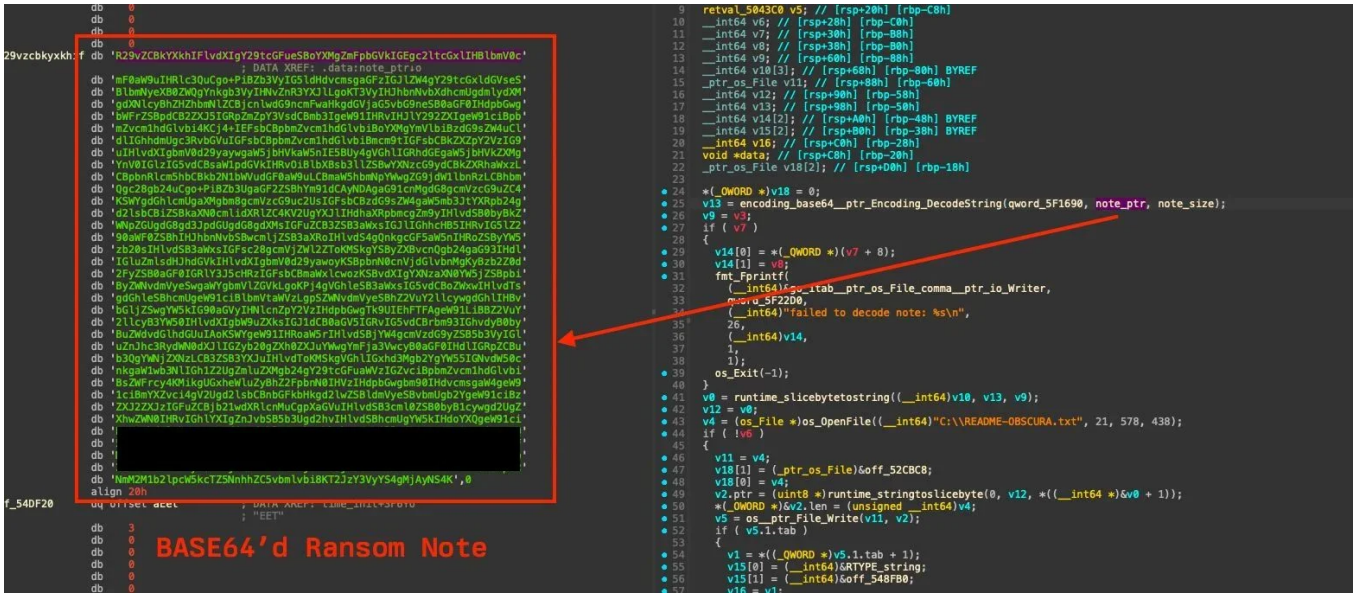


Figure 1: Base64'd ransom note being decoded

The main_windows_api_IsRunAsAdmin() function performs a Windows privilege check using two sequential Windows API calls to determine if the current process possesses administrative rights. The function first calls AllocateAndInitializeSid() to create a Security Identifier for the local Administrators group using SECURITY_BUILTIN_DOMAIN_RID (32) as the authority, DOMAIN_ALIAS_RID_ADMINS (544) as the subauthority, and an authority count of 2. Following successful SID creation, the function calls CheckTokenMembership() to verify if the current process token belongs to the Administrators group, returning a boolean value indicating administrative status. If either API call fails, the function returns descriptive error messages such as "AllocateAndInitializeSid failed:

%v" or "CheckTokenMembership failed: %v". When the privilege check determines the process lacks administrative rights, the ransomware prints "[!!!] user not admin. exit [!!!]" and immediately terminates execution. This represents a hard requirement with no bypass mechanism, as the ransomware requires administrative privileges to terminate system processes, delete volume shadow copies cmd.exe /c vssadmin delete shadows /all /quiet, and access system APIs necessary for domain detection and daemon process creation.

```
*(_OWORD *)v15 = 0;
v16 = 0;
*(_OWORD *)v17 = 0;
mw_sid_buffer = (_6_uint8 *)runtime_newobject((__int64)&_uint8::RTYPE);
*(_DWORD *)mw_sid_buffer = 0;
*(_WORD *)&(*mw_sid_buffer)[4] = 1280;
p__ptr_uint8 = (_ptr_uint8 *)runtime_newobject((__int64)&RTYPE__ptr_uint8);
p__11_uintptr = (_11_uintptr *)runtime_newobject((__int64)&RTYPE__11_uintptr);
(*p__11_uintptr)[0] = (uintptr)mw_sid_buffer; // Pointer to SID authority structure
(*p__11_uintptr)[1] = 2;                      // Authority count = 2 (for built-in domain)
(*p__11_uintptr)[2] = 32;                     // SECURITY_BUILTIN_DOMAIN_RID = 32
(*p__11_uintptr)[3] = 544;                    // DOMAIN_ALIAS_RID_ADMINS = 544 (Local Admins Group)
*(_OWORD *)&(*p__11_uintptr)[4] = 0;
*(_OWORD *)&(*p__11_uintptr)[6] = 0;
*(_OWORD *)&(*p__11_uintptr)[8] = 0;
(*p__11_uintptr)[10] = (uintptr)p__ptr_uint8;
v5 = syscall__ptr_LazyProc_Call(qword_5F2240, (__int64)p__11_uintptr, 11, 11);
v0 = v7;
if ( v5 )
```

Figure 2: Snippet of main_windows_api_IsRunAsAdmin that configures Windows security constants (2, 32, 544) to create Administrators group SID for privilege checking

After confirming administrative privileges, the ransomware gathers critical system information by calling GetSystemInfo() through the Windows API. It specifically extracts the dwNumberOfProcessors value, which indicates the number of CPU cores available on the system and is used for optimizing the threading strategy during the encryption phase. The system preparation phase continues with aggressive process termination targeting security and database applications that might interfere with the encryption process. The ransomware calls main_windows_api_KillProcesses(), which iterates through a predefined list of 120 target processes. The '*' found in some process names is used to indicate a wildcard for the string matching.

WinDefend	MsMpEng	MpCmdRun	CSFalconService	SentinelAgent
bdagent	McAfee	Avp	SymCorpUI	ccSvcHst
AMService	Emsisoft*	csrss_guard	traps*	cyserver
cytray	esensor*	elastic-endpoint*	f-secure*	fsav*
360tray	360sd	ksafe	avguard	avgnt
avast*	Crowdstrike*	falcon-sensor	glasswire*	ZoneAlarm
comodo*	Veeam*	VeeamTransportSvc	VeeamBackupSvc	AcrSch2Svc
Afcdpsrv	AcronisAgent	AcronsiBackupAgent	Altaro*	Nakivo*
Iperius*	MacriumService	EaseUS*	CrashPlanService	veritas*
NetBackup*	BackupExec	BEDatabase	BETracker	CommVault*
Cvd	Galaxy*	Snapman	StorageCraft*	druva*
rubrik*	synmedia*	cloudberry*	Dbagent	Datto*
SIRAgent	MSSQL*	SQLSERVERAGENT	SQLWriter	SQLBrowser
OracleService*	OracleVSSWriter	OracleXETNSListener	postgresql*	pg_ctl

mysql	mysqld	MariaDB	mariadb	percona*
ccbackup*	cbrestore*	ABBService	Splunkd	SplunkForwarder
ossec*	wazuh*	agent_m*	Zabbix*	nagios
Nrpe	prtg*	SolarWinds*	greylog*	Nxlog
Winlogon	EventLog	Sysmon*	VMwareHostd	VMwareAuthdServic
VMwareNatService	VMwareUSBArbZService	vmware-hostd	VBoxSDS	VBoxHeadless
VBox*	vmms	Vmicheartbeat	Vmickvpexchange	Vmicrdv
vmicshutdown	com.docker.service	gitlab-runner	jenkins*	TeamCity*
bamboo*	octopus*	rundeck*	ansible*	salt-minion
ActiveBackup*	Syno*	SynologyDrive	SynologyQuickConnect	

When a process name matches the target pattern above, the function executes the termination sequence by calling OpenProcess(PROCESS_TERMINATE, FALSE, processID) to obtain a handle to the target process with termination privileges. If the handle is successfully obtained, it calls TerminateProcess(process_handle, 1) to forcefully terminate the process with exit code 1 and prints a success message showing the process ID and name in the format “[+] killed pid %d (%s)”. If termination fails, the function returns an error message stating “failed to terminate process” but continues to kill other target processes.

The ransomware uses the Windows API DsRoleGetPrimaryDomainInformation to determine the computer's role in a domain. This is done in the main_windows_api_GetPCRole() function, which maps Windows domain roles to internal values.

Regardless of the detected domain role, each branch executes the same sequence of loading a role-specific string message and displaying corresponding status messages before immediately proceeding to the daemon creation phase. These messages suggest intended network propagation capabilities that were either never fully implemented or represent incomplete development, as the actual code contains no lateral movement functionality beyond the local encryption routine.

- Standalone PC: displays [+] detect standalone pc. indicating the system is not connected to a domain
- PC in Domain: shows [+] detect pc in domain. run transfer to dc. suggesting transfer to domain controllers
- Backup Domain Controller: shows [+] detect BDC. run transfer to PDC., implying propagation to the primary domain controller
- Primary Domain Controller: displays [+] detect PDC. run transfer to all pc in domain. indicating spread to all domain computers

There are a few encryption strategies the binary chooses from: EncryptFull or EncryptPart. Both of those functions make use of the encryptFileRange() function with different arguments.

They have a peer public key (Curve25519) and during encryption will generate an ephemeral private key using main_windows_api_generateEphemeralKeyPair(). These are used to generate the XChaCha20 key which is later used for file encryption. To accomplish this they use scalar multiplication (X25519) between the private key and their public key to generate a 32 byte shared secret. That shared secret along with a 24 byte random nonce are used as the parameters for the ChaCha file encryption. Before writing the encrypted file back to disk they append a 64 byte footer which is comprised of:

- OBSCURA!

- 32 byte public key
- 24 byte nonce

Figure 3: Sample of the encrypted file

Since they have the peer private key, they can use this footer to rederive the ChaCha20 key that was used to encrypt the file.

The Obscura ransomware implements a file filtering mechanism designed to maximize damage to user data while preserving system functionality.

The filtering system operates through the main_hasExcludedExtension() function, which performs case-insensitive extension matching against a hardcoded exclusion list. The function extracts file extension and compares against 15 predefined extensions:

System Executables and Libraries:

- .exe - Executable applications
- .dll - Dynamic Link Libraries
- .msi - Microsoft Installer packages
- .sys - System driver files

Boot and firmware components:

- .efi - UEFI firmware files
- .boot - Boot configuration files
- .iso - ISO disc image files

- .rom - ROM firmware files
- .bin - Binary system files

System configuration and utilities:

- .ini - Configuration files
- .cfg - Configuration files
- .lnk - Windows shortcut files
- .hosts - Network configuration files
- .swapfile - Windows virtual memory files

Ransomware self-protection:

- .obscura - encrypted files with ransomware extension

Obscura and other new ransomware variants

Obscura is one of several newer ransomware variants that Huntress has seen popping up in recent months, including Crux ransomware and Cephalus ransomware. This could be due to several factors. Threat actors continually rebrand and roll out new ransomware variants after law enforcement disruptions impact the ecosystem. Additionally, as our customer base continues to grow, we continue to gain more visibility into more ransomware variants.

Regardless, what was presented in this post is just one means for deploying ransomware. Organizations should monitor their domain controllers closely and look for the addition of new files, as well as the modification of existing files, including GPOs. Administrators should also monitor domain controllers, as well as other endpoints (servers, workstations) for unusual or suspicious access.

IOCs

Indicator	Description
[company name].exe	
sha256:	Ransomware executable
c00a2d757349bfff4d7e0665446101d2ab46a1734308cb3704f93d20dc7aac23	
README_Obscura.txt	Ransom note (contents below)
C:\WINDOWS\sysvol\sysvol\[domain].local\scripts\	Threat actor ops folder
DESKTOP-XNBSHKJ2	Possible threat actor workstation name

Sign Up for Huntress Updates

Get insider access to Huntress tradecraft, killer events, and the freshest blog updates.

Privacy • Terms

By submitting this form, you accept our Terms of Service & Privacy Policy

Thank you! Your submission has been received!

Oops! Something went wrong while submitting the form.