Yurei & The Ghost of Open Source Ransomware

⋮ 9/12/2025

Key Points

- First observed on September 5, Yurei is a newly emerged ransomware group that targeted a Sri Lankan food manufacturing company as its first leaked victim. The group follows a double-extortion model: they encrypt the victim's files and exfiltrate sensitive data, and then demand a ransom payment to decrypt and refrain from publishingthe stolen information.
- Check Point Research (CPR) determined that Yurei's ransomware is derived with only minor modifications from Prince-Ransomware, an open-source ransomware family written in Go. This highlights how open-source

malware significantly lowers the barrier to entry for cybercriminals, enabling even less-skilled threat actors to launch ransomware operations.

- Yurei's ransomware contains a flaw that may allow partial recovery through Shadow Copies, but the group primarily relies on data-theft-based extortion. As they stated on their blog, the fear and implications of data leakage are their main pressure point to get victims to pay the ransom.
- Since the first victim was listed on September 5, the number of victims has risen to three so far, pointing to a fast-growing operation.
- The investigation revealed hints that the threat actor's origins may be in Morocco.

Yurei Ransomware

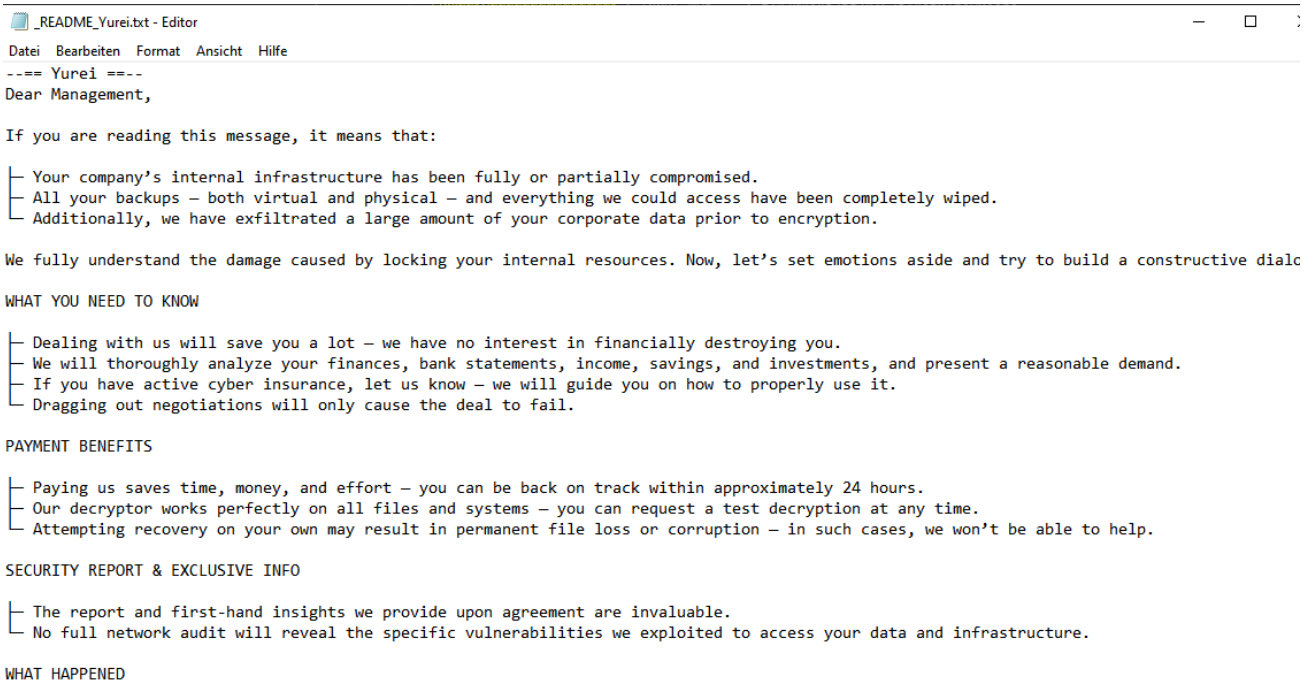
Check Point Research discovered a new ransomware group on September 5. The group calls themselves Yurei (a sort of spirit in Japanese folklore), and initially listed one victim, a Sri Lankan food manufacturing company, on their darknet blog. These blogs are used by ransomware groups to list their victims, show proofs of compromise such as screenshots of internal documents, and to provide a secure chat interface where the victim can negotiate with the operators. In the first few days of the operation, two new victims were listed, one from India and one from Nigeria, making a total of three as of September 9.

Figure 1 — Yurei ransomware site on September 5.

The Yurei ransomware is written in the Go programming language. While malware in Go is not uncommon, it still provides a challenge for some Antivirus vendors to detect. Combined with an easier development experience than C or C++ and the ability to cross-compile to different platforms, Go continues to be an attractive choice for malware developers.

In this case, the threat actor made the mistake of not stripping symbols from the binary. Therefore, function and module names were preserved, through which it becomes clear that Yurei's ransomware is largely based on an open-source ransomware named Prince-Ransomware (currently only available as a reupload on GitHub), with only minor modifications. This same ransomware codebase was already used in campaigns by other actors, such as in the case of CrazyHunter.

The ransom note is dropped as `_README_Yurei.txt` and instructs the victim to visit their site and enter their chat using a provided access token for further negotiation. Upon payment, the threat actor claims to provide a decryption tool as well as a report of the vulnerabilities exploited to compromise the environment, akin to a report of a penetration test.



_README_Yurei.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
--== Yurei ==--
Dear Management,

If you are reading this message, it means that:

├ Your company's internal infrastructure has been fully or partially compromised.
├ All your backups – both virtual and physical – and everything we could access have been completely wiped.
└ Additionally, we have exfiltrated a large amount of your corporate data prior to encryption.

We fully understand the damage caused by locking your internal resources. Now, let's set emotions aside and try to build a constructive dialo

WHAT YOU NEED TO KNOW

├ Dealing with us will save you a lot – we have no interest in financially destroying you.
├ We will thoroughly analyze your finances, bank statements, income, savings, and investments, and present a reasonable demand.
├ If you have active cyber insurance, let us know – we will guide you on how to properly use it.
└ Dragging out negotiations will only cause the deal to fail.

PAYMENT BENEFITS

├ Paying us saves time, money, and effort – you can be back on track within approximately 24 hours.
├ Our decryptor works perfectly on all files and systems – you can request a test decryption at any time.
└ Attempting recovery on your own may result in permanent file loss or corruption – in such cases, we won't be able to help.

SECURITY REPORT & EXCLUSIVE INFO

├ The report and first-hand insights we provide upon agreement are invaluable.
└ No full network audit will reveal the specific vulnerabilities we exploited to access your data and infrastructure.

WHAT HAPPENED
```

Figure 2 — Ransom note.

The victim is provided with the negotiation .onion page, where the victim and threat actors can communicate and negotiate the price of decryption and deletion of the stolen data as well as assurances against publishing corporate files.

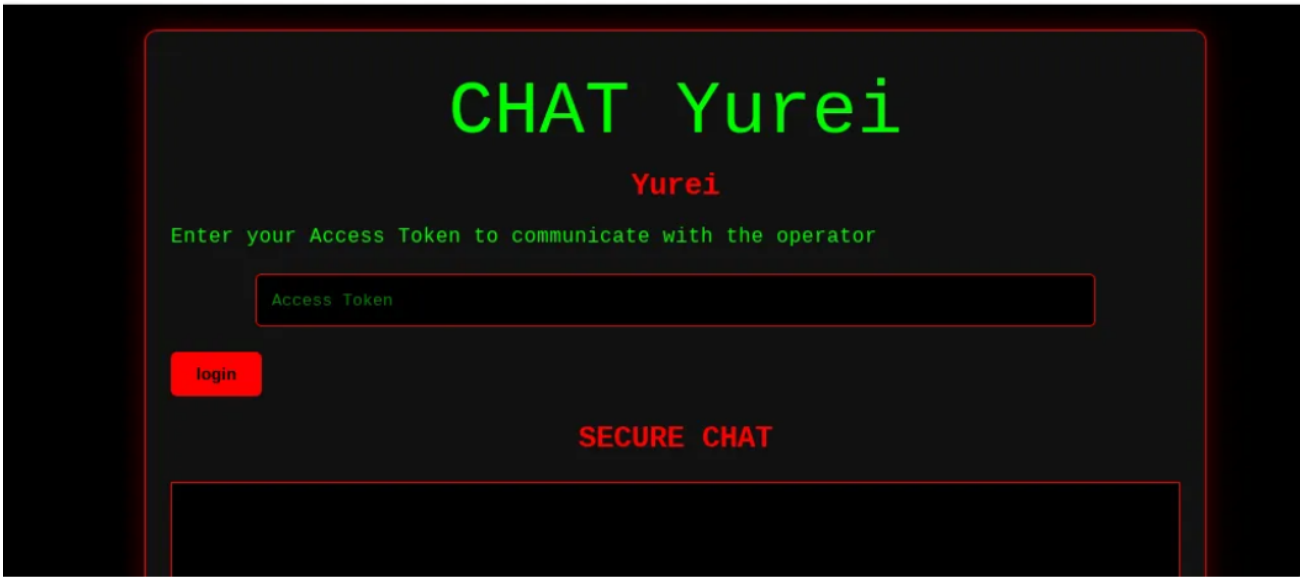


Figure 3 — Yurei chat interface.

Technical Analysis

As stated previously, Yurei is an offshoot of the open-source Prince-Ransomware, written in Go, with minor modifications. On a high level, the ransomware takes the following actions:

- Enumerates all drives
- For each drive in parallel, it encrypts files and adds the `.Yurei` extension
- Attempts to set a wallpaper
- Waits and monitors for newly attached network drives to then encrypt

Encryption

Files are encrypted using the ChaCha20 algorithm and are appended the `.Yurei` extension. The ransomware generates a random ChaCha20 key and a random nonce per file and then encrypts both with ECIES using the attacker's public key. The encrypted files then store the encrypted key, nonce, and file content, separated by the `||` characters:

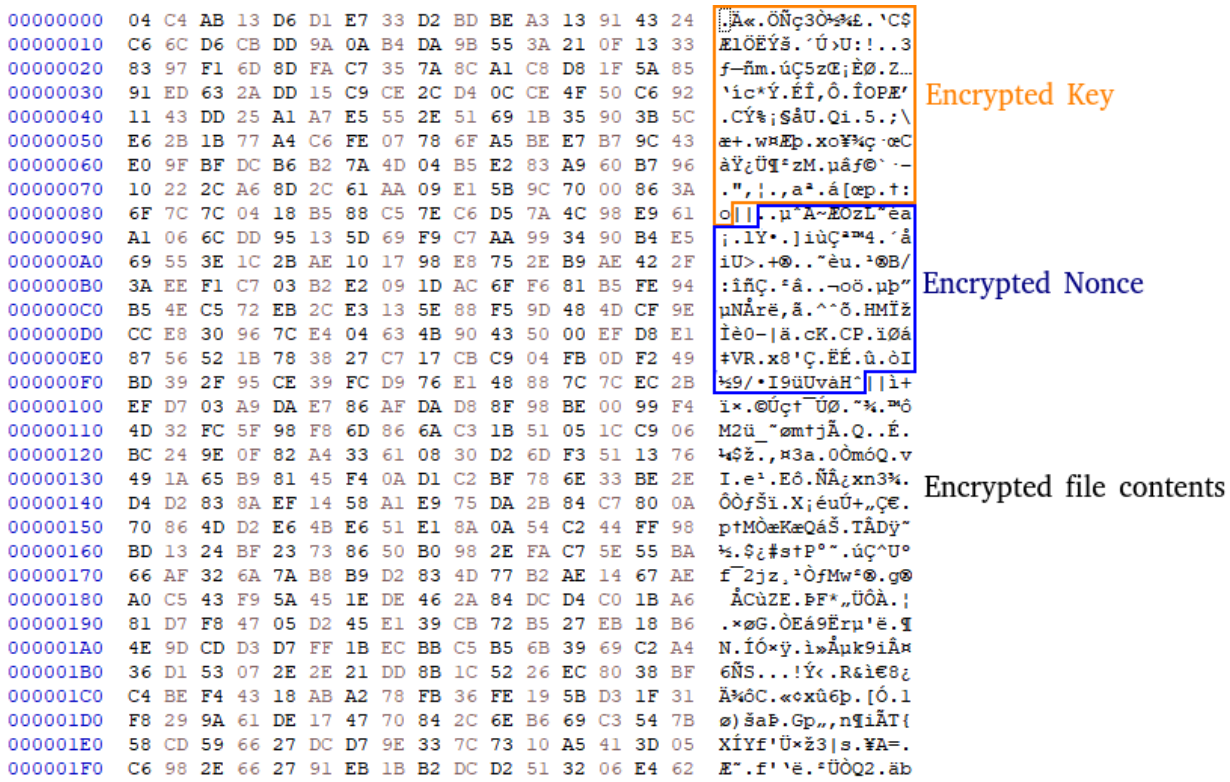


Figure 4 — Encrypted file structure.

Open Source Origin

The authors of the ransomware did not implement any anti-analysis features, such as obfuscation, but instead even included symbols in the shipped binary.

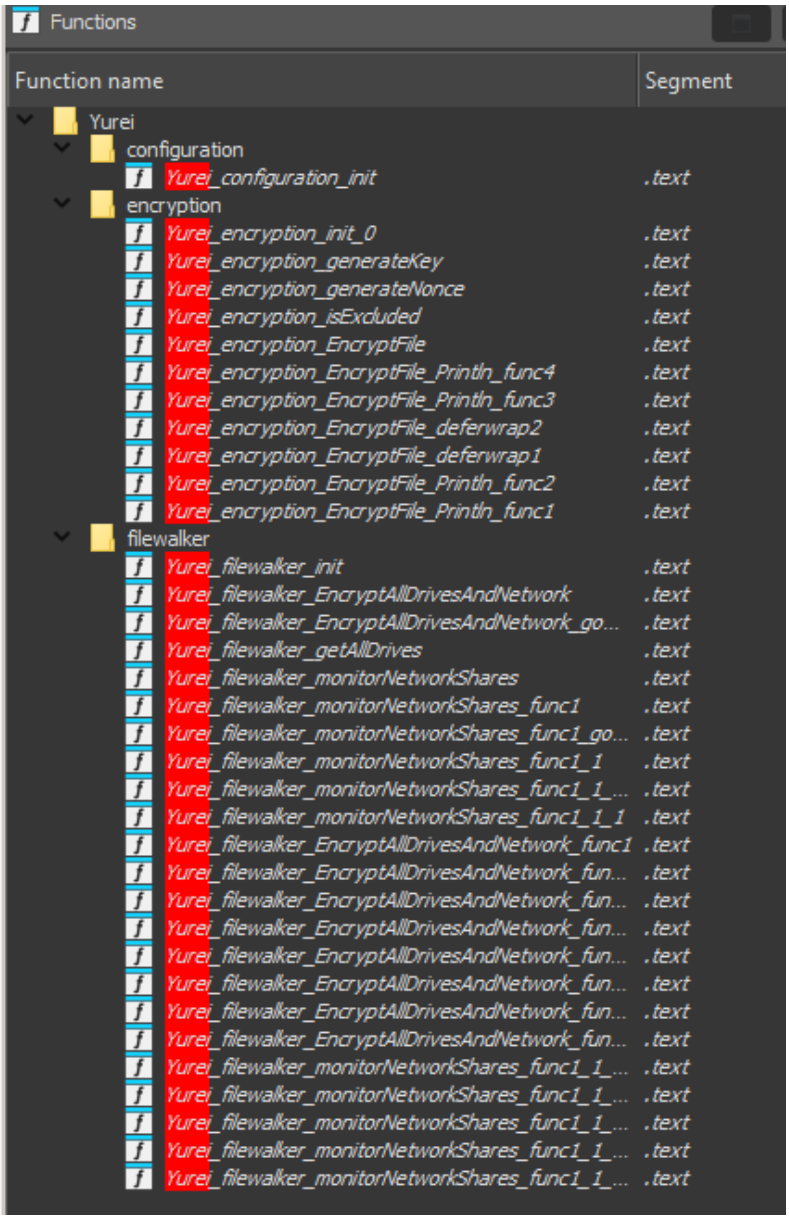


Figure 5 — Yurei modules and functions.

The same module names, `filewalker`, `encryption` and `configuration` are also used by the Open Source Prince-Ransomware:

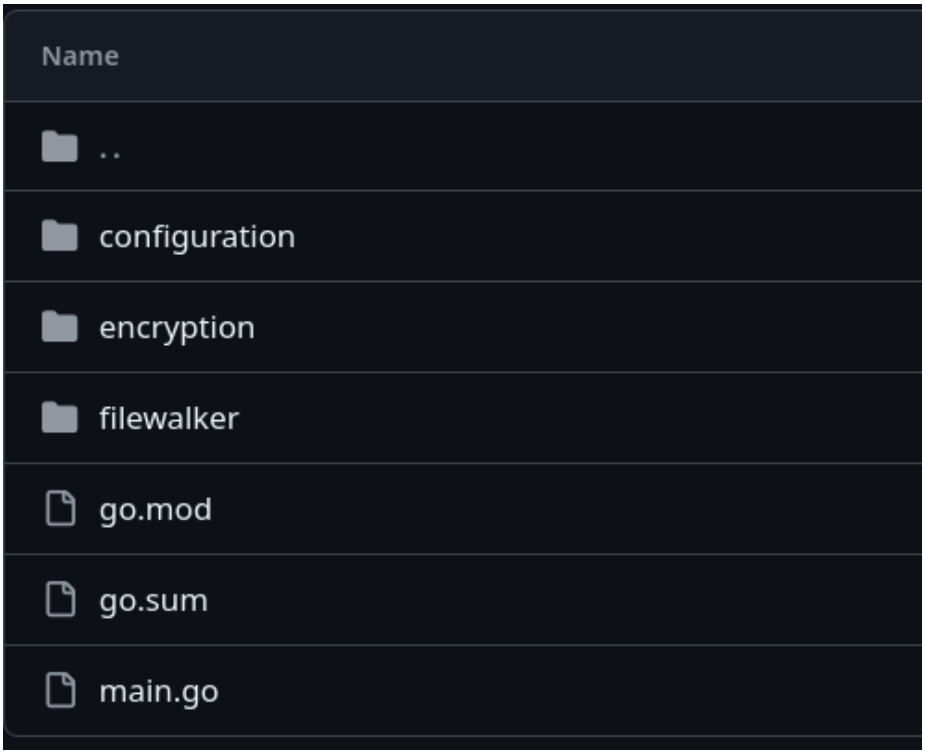


Figure 6 — Prince-Ransomware source files on GitHub.

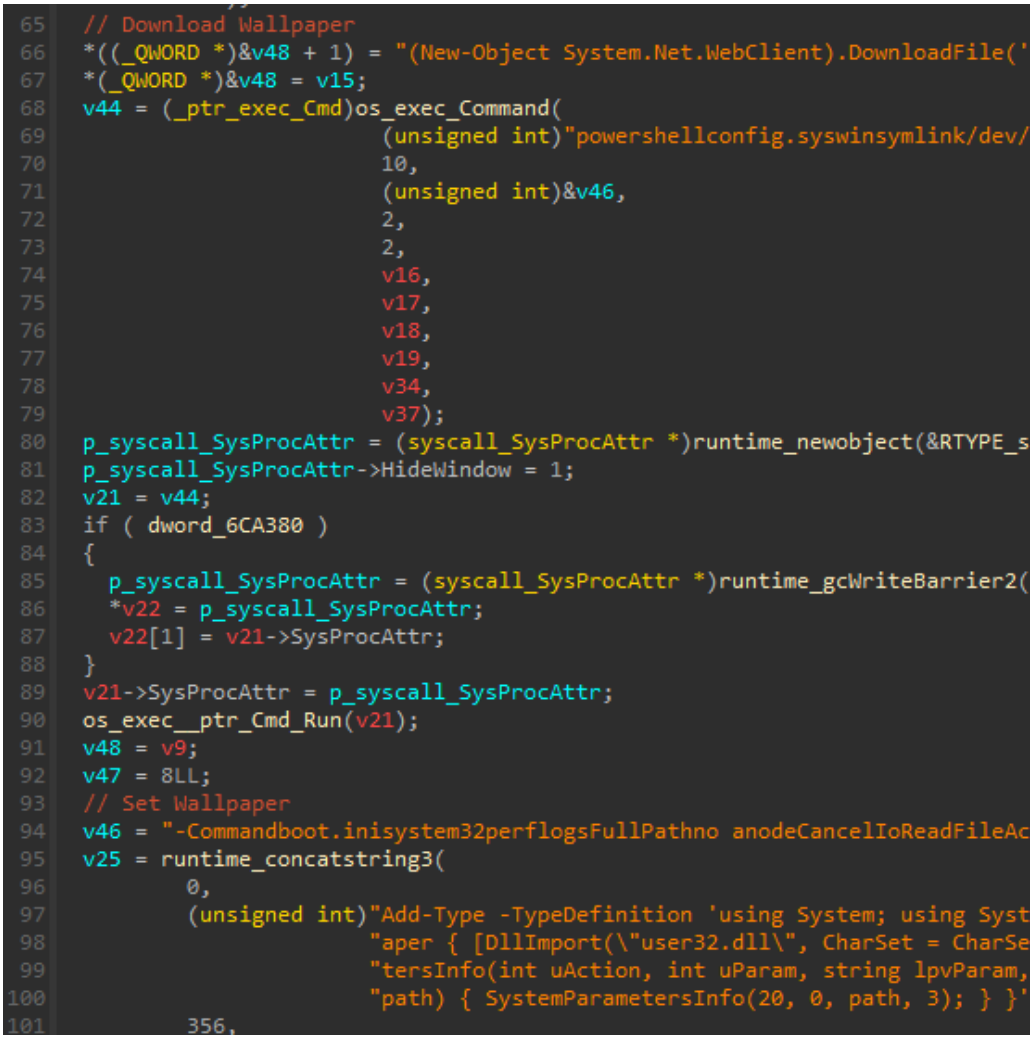
Further inspection of the code that changes the wallpaper confirms the suspicion that Yurei is based on Prince-Ransomware.

```
func setWallpaper() {
    filePath := filepath.Join(os.Getenv("TEMP"), "Wallpaper.png")

    // PowerShell command to download the image
    downloadCmd := exec.Command("powershell", "-Command", `(New-Object System.Net.WebClient).DownloadFile('`+Configuration.WallpaperURL+`', '`+filePath+`')`)
    downloadCmd.SysProcAttr = &syscall.SysProcAttr{HideWindow: true}
    err := downloadCmd.Run()
    if err != nil {
        return
    }

    // PowerShell command to set the wallpaper
    setWallpaperCmd := exec.Command("powershell", "-Command", `Add-Type -TypeDefinition 'using System; using System.Runtime.InteropServices; public class Wallpaper { [DllImport("user32.dll"
    setWallpaperCmd.SysProcAttr = &syscall.SysProcAttr{HideWindow: true}
    err = setWallpaperCmd.Run()
    if err != nil {
        return
    }
}
```
Figure 7 — Prince-Ransomware code to set the wallpaper.

Yurei carries over an unaltered set of the same PowerShell commands.



```
// Download Wallpaper
*((_QWORD *)&v48 + 1) = "(New-Object System.Net.WebClient).DownloadFile('
*(_QWORD *)&v48 = v15;
v44 = (_ptr_exec_Cmd)os_exec_Command(
                       (unsigned int)"powershellconfig.syswinsymlink/dev/
                       10,
                       (unsigned int)&v46,
                       2,
                       2,
                       v16,
                       v17,
                       v18,
                       v19,
                       v34,
                       v37);
p_syscall_SysProcAttr = (syscall_SysProcAttr *)runtime_newobject(&RTYPE_s
p_syscall_SysProcAttr->HideWindow = 1;
v21 = v44;
if ( dword_6CA380 )
{
  p_syscall_SysProcAttr = (syscall_SysProcAttr *)runtime_gcWriteBarrier2(
  *v22 = p_syscall_SysProcAttr;
  v22[1] = v21->SysProcAttr;
}
v21->SysProcAttr = p_syscall_SysProcAttr;
os_exec__ptr_Cmd_Run(v21);
v48 = v9;
v47 = 8LL;
// Set Wallpaper
v46 = "-Commandboot.inisystem32perflogsFullPathno anodeCancelIoReadFileAc
v25 = runtime_concatstring3(
        0,
        (unsigned int)"Add-Type -TypeDefinition 'using System; using Syst
                      "aper { [DllImport(\"user32.dll\", CharSet = CharSe
                      "tersInfo(int uAction, int uParam, string lpvParam,
                      "path) { SystemParametersInfo(20, 0, path, 3); } }'
        356,
```

Figure 8 — Yurei code to set the wallpaper.

The first command is supposed to download a wallpaper from a remote URL and save it as `Wallpaper.png` in the `%TEMP%` directory. The next command then compiles a .NET assembly to call `SystemParametersInfo` with `SPI_SETDESKWALLPAPER` to set the current wallpaper.

Usually, ransomware sets the wallpaper to a sort of Logo or ransom note. Interestingly, the Yurei developers did not supply a wallpaper to be downloaded. Therefore, the malware runs a download command via PowerShell that errors out due to a missing URL. As the wallpaper is set to a non-existing file in the subsequent PowerShell command, Windows falls back to setting the background to a single color, e.g., black.

Examining the builder's source code on GitHub reveals that the binary is shipped with symbols because the builder does not set the appropriate linker flags to strip them from the binary. The threat actors likely did not touch the builder code at all.

```
ldflags := fmt.Sprintf("-H=windowsgui -s -w -X 'Prince-Ransomware/configuration.PublicKey=%s'", pubKeyHex)
cmd := exec.Command("cmd", "/C", "go", "build", "-ldflags", ldflags, "-o", "../Prince-Built.exe")
cmd.Stdout = os.Stdout
cmd.Stderr = os.Stderr
err = cmd.Run()
if err != nil {
    return fmt.Errorf("build failed: %w", err)
}

fmt.Printf("Encryptor compiled successfully in %v\n", time.Since(startTime))
```

Figure 9 — Prince-Ransomware linker flags.

Yurei's modifications

While most of the source code was not modified, the code to enumerate which files and drives to encrypt differs slightly from the publicly available version on GitHub. While the original repository uses single-threaded encryption, Yurei makes use of goroutines, Go's concurrency mechanism, to encrypt each drive concurrently:

```
// Iterate over drives
for ( i = v42; i > 0; i = v42 - 1 )
{
  v42 = i;
  v44 = v26;
  v40 = v26[1];
  v43 = *v26;
  sync__ptr_WaitGroup_Add(&waitgroup, 1, v43, v46, v21, v22, v23, v24, v25);
  runtime_chansend1(qword_683E60, &v40);
  v28 = runtime_newobject(&unk_53F6E0);
  v33 = Yurei_filewalker_EncryptAllDrivesAndNetwork_gowrap1;
  *v28 = Yurei_filewalker_EncryptAllDrivesAndNetwork_gowrap1;
  v28[1] = &off_561E88;
  v21 = v40;
  v28[3] = v40;
  if ( dword_6CA380 )
  {
    v28 = runtime_gcWriteBarrier1();
    v20 = v43;
    *v32 = v43;
  }
  else
  {
    v20 = v43;
  }
  v28[2] = v20;
  // Initialize goroutine
  runtime_newproc(v28, v20, v33, v46, v21, v29, v30, v31, v32);
  v26 = v44 + 2;
}
sync__ptr_WaitGroup_Wait(&waitgroup);
return runtime_newproc(&monitor_network_shares_routine, v20, v34, v46, v21, v35, v36, v37, v38);
```

Figure 10 — Yurei goroutines.

Once the encryption is finished, the malware enters a new routine which continuously monitors for new network drives to add to the encryption queue:

```
while ( 1 )
{
  v13[1] = 2LL;
  v13[0] = &base_path;                          // "\\"
  v9 = v13;
  for ( i = 0LL; i <= 0; i = v12 + 1 )
  {
    v12 = i;
    v14 = v9;
    a2 = v9[1];
    // Walk path and append to workgroup if new drive is found
    path_filepath_Walk(*v9, a2, &callback_function, a4, a5, a6, a7, a8, a9);
    v9 = v14 + 2;
  }
  // Sleep for ~2 seconds
  time_Sleep(0x7E11D600, a2, a3, a4, a5, a6, a7, a8, a9, v11);
}
```

Figure 11 — Yurei `monitorNetworkShares` routine.

While there is no evidence that this was AI-assisted development, these types of modifications on top of existing codebases can easily be performed even by lower-skilled developers, using simple prompts on Large Language Models (LLMs).

Shadow Copy Recovery

Although the threat actor modified the codebase, the Yurei ransomware still has a major flaw: It does not delete existing Shadow copies. Shadow copies are backup snapshots of files or entire volumes that, if enabled, are generated by the Volume Shadow Copy Service (VSS). Ransomware usually targets and deletes these copies to block victims from using Windows' built-in recovery options.

As Yurei does not include this functionality, if Shadow Copies are enabled, the Victim can **restore their files to a previous snapshot** without having to negotiate with Yurei. This oversight again shows the lack of sophistication this threat actor employs in its operation. As a result, activating VSS and continuously taking snapshots of systems is highly recommended as a protective measure against threats such as Yurei. However, ransomware groups are increasingly shifting to data-theft-based extortion, so this only aids in operational recovery but does not protect against extortion. As Yurei states on their blog, the main pressure point for victims paying the ransom is the threat of data leakage. This has major implications for businesses, and in the case of e.g. Midcity Marketing, can also affect food security and state supply chains.

Figure 12 — Yurei on the implications of a possible Leak.

Tracing the Threat Actor

Looking at the samples on VirusTotal shows that all samples were first submitted from Morocco. One sample did not include a ticket ID, indicating that this could be a test build, possibly uploaded by the developer themselves. It is possible that the threat actor may have used VirusTotal as a means to test the detection rate of his ransomware, once again showing that we are dealing with a low-skilled actor.

When we analyzed the HTML source code of the .onion page, we found a comment in Arabic:

```
  // ؟
  document.addEventListener('contextmenu',e=>e.preventDefault());
  document.addEventListener('keydown',function(e){
    if(e.keyCode===123||(e.ctrlKey&&e.shiftKey&&e.keyCode===73)||(e.ctrlKey&&e.keyCode===85)){
      e.preventDefault();
    }
  });
</script>
</body>
</html>
```

Figure 13 — Arabic comment inside HTML code from the .onion page.

The path artifacts from the available samples all list the local path as `D:\satanlockv2\*`, indicating possible ties to the SatanLockv2 ransomware. Looking at other samples of SatanLockv2, they were first uploaded from Morocco and based on Prince-Ransomware as well.

As a result, we assess with low confidence that the threat actor is based in Morocco.

Conclusion

Yurei demonstrates how easily threat actors can weaponize open open- source ransomware projects with only minimal modifications, enabling low-level threat actors to enter the ransomware business without the necessary development skills or even investing much effort. By reusing the codebase of Prince-Ransomware, the group managed to launch operations quickly but also inherited its flaws, most notably the failure to remove Volume Shadow Copies. This oversight enables partial recovery in environments where VSS is enabled. As ransomware groups increasingly focus on data-theft-based extortion, however, this flaw, does not have a critical impact on the ransomware operation's success.

While open-source malware is a threat, it also gives defenders opportunities to detect and mitigate these variations. However, Yurei succeeded in running their operation on several victims, which shows that even low-effort operations can still lead to success

Protections

Check Point Threat Emulation and Harmony Endpoint provide comprehensive coverage of attack tactics, file types, and operating systems and protect against the attacks and threats described in this report.

Indicators of Compromise

Description	Value
Onion Page	fewcriet5rhoy66k6c4cyvb2pqrblxtx4mekj3s5l4jjt4t4kn4vheyd.onion

Description	Value
Yurei Ransomware	49c720758b8a87e42829ffb38a0d7fe2a8c36dc3007abfabbea76155185d2902 4f88d3977a24fb160fc3ba69821287a197ae9b04493d705dc2fe939442ba6461 1ea37e077e6b2463b8440065d5110377e2b4b4283ce9849ac5efad6d664a8e9e 10700ee5caad40e74809921e11b7e3f2330521266c822ca4d21e14b22ef08e1d 89a54d3a38d2364784368a40ab228403f1f1c1926892fe8355aa29d00eb36819 f5e122b60390bdcc1a17a24cce0cbca68475ad5abee6b211b5be2dea966c2634 0303f89829763e734b1f9d4f46671e59bfaa1be5d8ec84d35a203efbfcb9bb15
SatanLockV2 Ransomware	afa927ca549aaba66867f21fc4a5d653884c349f8736ecc5be3620577cf9981f d2539173bdc81503bf1b842a21d9599948e957cadc76a283a52f5849323d8e04

Ransom Note

Plain text

Copy to clipboard

Open code in new window

EnlighterJS 3 Syntax Highlighter

--== Yurei ==--

Dear Management,

If you are reading this message, it means that:

├─ Your company’s internal infrastructure has been fully or partially compromised.

├─ All your backups — both virtual and physical — and everything we could access have been completely wiped.

└─ Additionally, we have exfiltrated a large amount of your corporate data prior to encryption.

We fully understand the damage caused by locking your internal resources. Now, let’s set emotions aside and try to build a constructive dialogue.

WHAT YOU NEED TO KNOW

├─ Dealing with us will save you a lot — we have no interest in financially destroying you.

├─ We will thoroughly analyze your finances, bank statements, income, savings, and investments, and present a reasonable demand.

├─ If you have active cyber insurance, let us know — we will guide you on how to properly use it.

└─ Dragging out negotiations will only cause the deal to fail.

PAYMENT BENEFITS

├─ Paying us saves time, money, and effort — you can be back on track within approximately 24 hours.

├─ Our decryptor works perfectly on all files and systems — you can request a test decryption at any time.

└─ Attempting recovery on your own may result in permanent file loss or corruption — in such cases, we won't be able to help.

SECURITY REPORT & EXCLUSIVE INFO

├─ The report and first-hand insights we provide upon agreement are invaluable.

└─ No full network audit will reveal the specific vulnerabilities we exploited to access your data and infrastructure.

WHAT HAPPENED

├─ Your network infrastructure has been compromised.

├─ Critical data has been exfiltrated.

└─ Files have been encrypted.

WHAT YOU SHOULD NOT DO

├─ Do NOT rename, modify, or delete encrypted files.

├─ Do NOT shut down your system or run antivirus software — this may cause irreversible damage.

└─ Do NOT waste time with data recovery companies — they cannot help you.

VALUABLE DATA WE USUALLY STEAL

├─ Databases, legal documents, and personal information

├─ Audit reports, SQL databases

├─ Financial documents: statements, invoices, accounting data

├─ Work files and corporate communications

├─ Any backup solutions

└─ Confidential documents

TO DO LIST (Best Practices)

├─ Contact us as soon as possible via our live chat (only).

├─ Purchase our decryption tool — there is no other way to recover your data.

├─ Avoid third-party negotiators or recovery services.

└─ Do not attempt to use public decryption tools — you risk permanent data loss.

RESPONSIBILITY

├─ Violating the terms of this offer will result in:

│ - Deletion of your decryption keys

│ - Immediate sale or public disclosure of your leaked data

│ - Notification of regulatory agencies, competitors, and clients

---

CHAT: Yurei

CHAT:hxxp://fewcriet5rhoy66k6c4cyvb2pqrblxtx4mekj3s5l4jjt4t4kn4vheyd.onion/chat/<VICTIM-GUID/chat.php

Your Ticket ID: <TICKET-ID>

Blog:hxxp://fewcriet5rhoy66k6c4cyvb2pqrblxtx4mekj3s5l4jjt4t4kn4vheyd.onion

---

Thank you for your attention.

Important Notes:

- Renaming, copying, or moving encrypted files may break the cipher and make decryption impossible.

- Using third-party recovery tools can irreversibly damage encrypted files.

- Shutting down or restarting the system may cause boot or recovery errors and further damage the encrypted data.

--== Yurei ==-- Dear Management, If you are reading this message, it means that: ├─ Your company’s internal infrastructure has been fully or partially compromised. ├─ All your backups — both virtual and physical — and everything we could access have been completely wiped. └─ Additionally, we have exfiltrated a large amount of your corporate data prior to encryption. We fully understand the damage caused by locking your internal resources. Now, let’s set emotions aside and try to build a constructive dialogue. WHAT YOU NEED TO KNOW ├─ Dealing with us will save you a lot — we have no interest in financially destroying you. ├─ We will thoroughly analyze your finances, bank statements, income, savings, and investments, and present a reasonable demand. ├─ If you have active cyber insurance, let us know — we will guide you on how to properly use it. └─ Dragging out negotiations will only cause the deal to fail. PAYMENT BENEFITS ├─ Paying us saves time, money, and effort — you can be back on track within approximately 24 hours. ├─ Our decryptor works perfectly on all files and systems — you can request a test decryption at any time. └─ Attempting recovery on your own may result in permanent file loss or corruption — in such cases, we won’t be able to help. SECURITY REPORT & EXCLUSIVE INFO ├─ The report and first-hand insights we provide upon agreement are invaluable. └─ No full network audit will reveal the specific vulnerabilities we exploited to access your data and infrastructure. WHAT HAPPENED ├─ Your network infrastructure has been compromised. ├─ Critical data has been exfiltrated. └─ Files have been encrypted. WHAT YOU SHOULD NOT DO ├─ Do NOT rename, modify, or delete encrypted files. ├─ Do NOT shut down your system or run antivirus software — this may cause irreversible damage. └─ Do NOT waste time with data recovery companies — they cannot help you. VALUABLE DATA WE USUALLY STEAL ├─ Databases, legal documents, and personal information ├─ Audit reports, SQL databases ├─ Financial documents: statements, invoices, accounting data ├─ Work files and corporate communications ├─ Any backup solutions └─ Confidential documents TO DO LIST (Best Practices) ├─ Contact us

as soon as possible via our live chat (only). ├─ Purchase our decryption tool — there is no other way to recover your data. ├─ Avoid third-party negotiators or recovery services. └─ Do not attempt to use public decryption tools — you risk permanent data loss. RESPONSIBILITY ├─ Violating the terms of this offer will result in: │ - Deletion of your decryption keys │ - Immediate sale or public disclosure of your leaked data │ - Notification of regulatory agencies, competitors, and clients --- **CHAT:** Yurei CHAT:hxxp://fewcriet5rhoy66k6c4cyvb2pqrblxtx4mekj3s5l4jjt4t4kn4vheyd.onion/chat/<VICTIM-GUID/chat.php Your Ticket ID: <TICKET-ID> Blog:hxxp://fewcriet5rhoy66k6c4cyvb2pqrblxtx4mekj3s5l4jjt4t4kn4vheyd.onion --- Thank you for your attention. --- **Important Notes:** - Renaming, copying, or moving encrypted files may break the cipher and make decryption impossible. - Using third-party recovery tools can irreversibly damage encrypted files. - Shutting down or restarting the system may cause boot or recovery errors and further damage the encrypted data.

```
---== Yurei ==--
Dear Management,

If you are reading this message, it means that:

├─ Your company’s internal infrastructure has been fully or partially compromised.
├─ All your backups — both virtual and physical — and everything we could access have been completely wiped.
└─ Additionally, we have exfiltrated a large amount of your corporate data prior to encryption.

We fully understand the damage caused by locking your internal resources. Now, let’s set emotions aside and try to build a constructive dialogue.

WHAT YOU NEED TO KNOW

├─ Dealing with us will save you a lot — we have no interest in financially destroying you.
├─ We will thoroughly analyze your finances, bank statements, income, savings, and investments, and present a reasonable demand.
├─ If you have active cyber insurance, let us know — we will guide you on how to properly use it.
└─ Dragging out negotiations will only cause the deal to fail.

PAYMENT BENEFITS

├─ Paying us saves time, money, and effort — you can be back on track within approximately 24 hours.
├─ Our decryptor works perfectly on all files and systems — you can request a test decryption at any time.
└─ Attempting recovery on your own may result in permanent file loss or corruption — in such cases, we won’t be able to help.

SECURITY REPORT & EXCLUSIVE INFO
```

├ The report and first-hand insights we provide upon agreement are invaluable.
└ No full network audit will reveal the specific vulnerabilities we exploited to access your data and infrastructure.

WHAT HAPPENED

├ Your network infrastructure has been compromised.
├ Critical data has been exfiltrated.
└ Files have been encrypted.

WHAT YOU SHOULD NOT DO

├ Do NOT rename, modify, or delete encrypted files.
├ Do NOT shut down your system or run antivirus software — this may cause irreversible damage.
└ Do NOT waste time with data recovery companies — they cannot help you.

VALUABLE DATA WE USUALLY STEAL

├ Databases, legal documents, and personal information
├ Audit reports, SQL databases
├ Financial documents: statements, invoices, accounting data
├ Work files and corporate communications
├ Any backup solutions
└ Confidential documents

TO DO LIST (Best Practices)

├ Contact us as soon as possible via our live chat (only).
├ Purchase our decryption tool — there is no other way to recover your data.
├ Avoid third-party negotiators or recovery services.
└ Do not attempt to use public decryption tools — you risk permanent data loss.

RESPONSIBILITY

├ Violating the terms of this offer will result in:
│ - Deletion of your decryption keys
│ - Immediate sale or public disclosure of your leaked data
│ - Notification of regulatory agencies, competitors, and clients

---

CHAT: Yurei
CHAT:hxxp://fewcriet5rhoy66k6c4cyvb2pqrblxtx4mekj3s5l4jjt4t4kn4vheyd.onion/chat/<VICTIM-

GUID/chat.php
Your Ticket ID: <TICKET-ID>
Blog:hxxp://fewcriet5rhoy66k6c4cyvb2pqrblxtx4mekj3s5l4jjt4t4kn4vheyd.onion
---

Thank you for your attention.

Important Notes:

- Renaming, copying, or moving encrypted files may break the cipher and make decryption impossible.
- Using third-party recovery tools can irreversibly damage encrypted files.
- Shutting down or restarting the system may cause boot or recovery errors and further damage the encrypted data.

GO UP
BACK TO ALL POSTS