A Case Study on Interlock Ransomware with Process Demonstration

Nikoloz Kurtanidze & James McGrath

Abstract

This paper outlines the tactics, tools, and motivations behind the Interlock Ransomware group, which emerged publicly in 2024 and has since been responsible for multiple high-profile attacks. Interlock leverages social engineering, Remote Access Tools (RATs), and off-the-shelf utilities like AnyDesk and Azure Storage Explorer to infiltrate networks, exfiltrate data, and encrypt systems for ransom. We walk through their full attack process, including fake browser updaters, LOLBins abuse, and custom encryption techniques. Additionally, we provide a demonstration of a simulated ransomware attack using a phishing campaign, RustDesk remote access, and Python programs we wrote for keylogging and encryption—all conducted in a safe, virtual lab environment. The goal was not to deploy real malware but to replicate the process as closely as possible within legal and resource constraints. We also touch on mitigation strategies, including findings from law enforcement initiatives like Operation Endgame. In the end, what's most concerning isn't the complexity of these attacks—it's how deceptively simple they are to pull off.

Introduction

The Interlock Ransomware group was first publicly discovered and reported on in September of 2024. Their prerogative is to target various big organizations, exploit their vulnerabilities, and hold those exploits ransom. "Interlock claims to target organizations' infrastructure by exploiting unaddressed vulnerabilities. It claims their actions are in part motivated by a desire to hold companies accountable for poor cybersecurity, in addition to monetary gain" [1]. Interlock runs their blog *Worldwide Secrets Blog*, which is where they post their data leaks if their ransoms are not paid. Interlock allegedly has ties to Rhysida Ransomware Group, which is a threat actor that operates as a ransomware-as-a-service (RaaS) group.

In this instance, Interlock attacked multiple targets but used the same process for all of them. The highest level of specificity that was given was that these targets were one of the following: healthcare organizations, technology companies, United States government entities, and European manufacturing organizations. The primary targets were the ones with the most exploits. Interlock was after sensitive data like intellectual property and/or confidential information. This information would be stolen and encrypted with ransomware. Then Interlock would threaten the release of this information to their aforementioned public blog *Worldwide Secrets Blog* unless their demanded ransom was paid.

Background

To start their attack, Interlock entered the target network by using social engineering techniques. The hackers were able to fool victims into downloading a fraudulent browser updater from compromised legal websites. From there, they again further exploited unaddressed vulnerabilities, weak access controls, and poor segmentation to escalate privileges and move laterally within networks. This attack was extremely complex, and clearly required a lot of preparation. On top of this, Interlock used a lot of pre-existing tools, methods, and libraries to conduct their attack(s).

According to a comprehensive review by [3], ransomware attacks have grown in sophistication through the adoption of stealthier payload delivery, targeted victim selection, and automated encryption techniques—all of which mirror Interlock's tactics. Additionally, [4] points out that modern ransomware operations increasingly blur the line between state-sponsored espionage and criminal enterprise, especially when attackers claim moral motives alongside financial ones, as Interlock does. Recent studies on Rhysida ransomware uncovered that attackers exploited predictable seeds in their cryptographic routines, enabling researchers to reconstruct encryption keys without ransom payments [5]. While Interlock's implementation has not been decrypted, the overlap in tools—such as the use of LibTomCrypt—suggests potential vulnerabilities of a similar nature. The broader evolution of ransomware—from crude scareware to enterprise-level threats—is documented in [6], emphasizing how attackers now mimic legitimate business operations and software pipelines to evade detection and improve monetization.

The main lesson to be learned here is that attackers are getting creative. What it boils down to is that Interlock exploited several security flaws during its operation. For example, victim systems that were found with outdated or unpatched software, weak RDP configurations, and a lack of robust password policies. The infiltration involved tricking the user into downloading an update for a browser they trust (Chrome). They downloaded this update installer, from a commonly-used and trusted website that happened to be compromised. Furthermore, the absence of two-factor authentication (2FA) for sensitive accounts and inadequate firewall and EDR measures further enabled the hackers to succeed in their infiltration.

To combat this, In May of 2024, law enforcement agencies in the U.S. and Europe coordinated to release a counter to these types of attacks, from these types of attackers. This is called *Operation Endgame.* According to the article *'Operation Endgame' Hits Malware Delivery Platforms*, To prevent malware infection through RATs, and delivery through droppers, companies should adopt proactive security protocols. Firstly, ensuring robust email security systems to block phishing attempts, which are common vectors for RATs and droppers. Secondly, organizations must prioritize endpoint detection and response (EDR) solutions to identify and mitigate malicious activity before it escalates. The article emphasizes the danger of cybercriminals using "paid ads on Google to trick people into installing malware disguised as popular free software" [2], which highlights the need to educate employees and users about verifying software sources and avoiding suspicious downloads. Education like this is the exact thing that would have prevented this series of attacks from Interlock. This would be a direct counter! Furthermore, maintaining up-to-date security patches and implementing strict access controls can limit vulnerabilities that attackers can exploit. These strategies, alongside coordinated global efforts like *Operation Endgame*, can disrupt the infrastructure supporting these cybercriminal tools and reduce their effectiveness.

Methodology

The location of the start piece of this attack occurs at endpoints through a fake browser updater. Interlock had compromised several commonly used website URLs. On these web pages, they would prompt the user to update their Chrome browser. The user would click this prompt and an executable would be downloaded. This fake browser updater executable was a Remote Access Tool (RAT) that would automatically execute a Powershell script. The script would download an actual “ChromeSetup.exe” as a decoy. Upon every user login, the RAT would run in the Windows StartUp folder. The hackers would further navigate these systems through a Remote Desktop Protocol control they established during all of this. Furthermore, they used tools like AnyDesk, PuTTY, and possibly LogMeIn to navigate within compromised systems. Finally, Interlock used Azure Storage Explorer and AzCopy to extract the sensitive information that they were targeting from networks, logical drives, and folders.

This attack, from the initial compromise to the deployment of the ransomware, was conducted within 17 days. For a majority of that time, the RAT stayed dormant, allowing the attackers to remotely connect and collect the information that they were targeting. To reiterate why these organizations were the targets. Interlock wanted to hold them accountable for poor cybersecurity practices and exploit their vulnerabilities for big payouts through ransoms.

Tools and Execution

The first tool that was used was a Fake Chrome Browser Updater, RAT, disguised as a legitimate application. This executable would run persistently every time the user logged in on the systems. It would then promptly gather system information and enable the attackers to execute PowerShell scripts. The RAT also deployed PowerShell commands to download additional malware, including a credential stealer and a keylogger.

The credential stealer targeted browser-stored credentials, key databases such as key4.db, and browsing histories. Interlock also used SQL queries to extract sensitive login information from the systems they compromised. Similarly, the keylogger captured keystrokes and stored logs locally on infected devices. The attackers executed it using rundll32.exe, a Windows tool that is a known example of "Living-off-the-Land Binaries" (LOLBins) abuse. Attacks like this that incorporate LOLBins abuse are difficult to detect since the whole premise of LOLBins abuse is that the attacker is manipulating natively-installed components on a Windows system.

Through RDP, the Interlock attackers maneuvered around the environments using Azure Storage Explorer and AzCopy to find their target files. From here they allegedly deployed endpoint detection and response (EDR) evasion tools, which may have included EDR uninstaller utilities or exploited vulnerable drivers to disable security measures. In addition to all of this, Interlock had some of its own custom-built tools ready to use. They deployed their Interlock ransomware binary, which encrypted files, appended a .interlock extension, and delivered ransom notes to victims. This ransomware utilized the LibTomCrypt library for encryption. They even utilized separate encryption techniques for Windows and Linux machines. Implementing Cipher Block Chaining (CBC) on Windows and CBC/RSA for Linux systems.

Attribution and Impact

Currently, nobody has been able to identify specifically who did this. Some firms and investigative teams had linked the attack to Interlock Ransomware Group. Where, again, Interlock is allegedly linked to an even bigger threat actor Rhysida Ransomware group. One of the leading pieces of evidence is the .interlock files. Even still, individuals have yet to be identified. Consequently, there have been no formal charges filed against the perpetrators. There was also no disclosure of what companies were held at ransom. Probably to avoid reputational damages. Financial damages have not been disclosed, but it is implied there will be money spent by the victims to hire new teams and team members to remediate their systems. Plus, all the money they lost during the operational downtime.

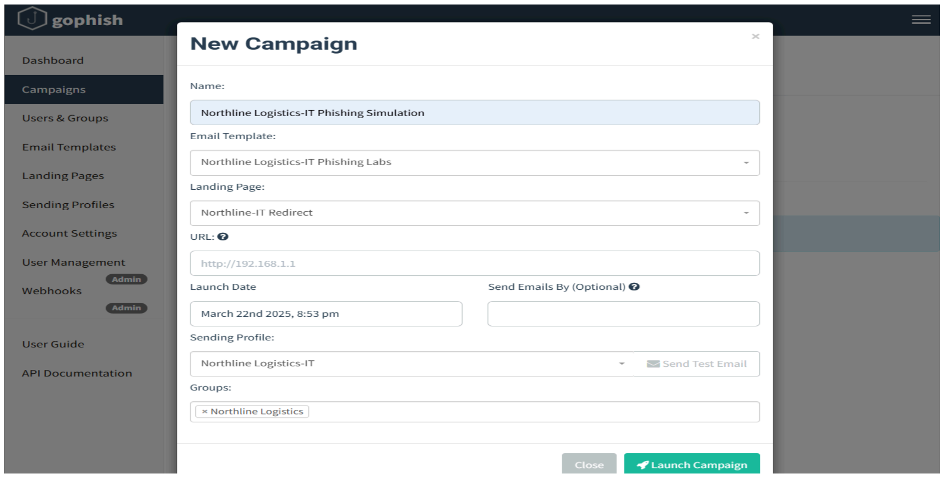
Demonstration

Our demonstration was highly experimental. The experimental settings were the following: my Windows 11 computer served as the host, attacker, and victim simultaneously. By host, I mean I had the computer host a Virtual Machine (VM) running Windows 10 Home. By attacker, I mean I used the host computer to attack the VM. By victim, I mean the VM.

There was no "data sampling," because the goal of this demonstration wasn't to extract data or simulate an advanced persistent threat. The goal was just to show how a ransomware attack like Rhysida's *could* be done—more specifically, to illustrate the process of a ransomware attack, from start to finish. The reason for that constraint is, well, two things: one, RATs and other Trojan horses are illegal to even code. And two, we had some resource constraints—namely, running a Windows 10 VM is horrifically slow, even on a decently powerful machine. We allocated as much RAM and CPU cores as we could to the VM without compromising the host machine, but it was still extremely slow, which didn't leave much flexibility for our process. Despite this, we still were able to simulate the process of a ransomware attack.

First, we set up the victim VM using VirtualBox and created a fake victim email account. Then, using GoPhish, we created a phishing campaign. The email pretended to be from the company (Northline Logistics)[1] IT Department, saying that to better accommodate remote and hybrid employees, the department was enabling Remote Desktop access on company laptops. The victim was then instructed to download RustDesk.

[1] Northline Logistics was supposed to be a fake company name. Upon checking, we realized it is, in fact, a real company, it was too late to go back and change such a small detail. This doesn't really affect anything, but we felt that it would be wise to clarify that.



New Campaign

Name: Northline Logistics-IT Phishing Simulation

Email Template: Northline Logistics-IT Phishing Labs

Landing Page: Northline-IT Redirect

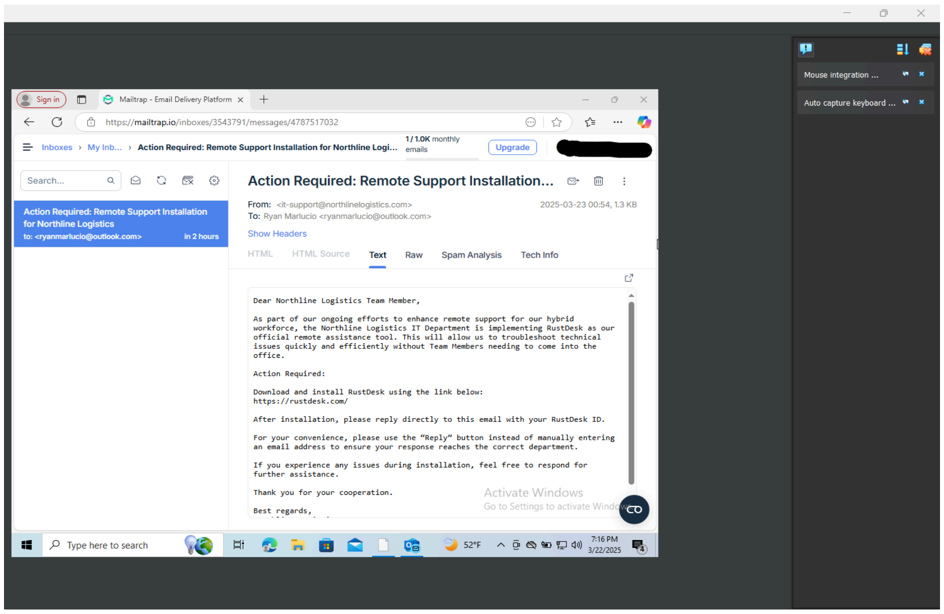
URL: http://192.168.1.1

Launch Date: March 22nd 2025, 8:53 pm

Send Emails By (Optional):

Sending Profile: Northline Logistics-IT

Groups: × Northline Logistics


Mouse integration ...
Auto capture keyboard ...
Sign in
Mailtrap - Email Delivery Platform
https://mailtrap.io/inboxes/3543791/messages/4787517032
Inboxes
My Inb...
Action Required: Remote Support Installation for Northline Logi...
1 / 1.0K monthly emails
Upgrade
Search...
HTML
HTML Source
Text
Raw
Spam Analysis
Tech Info
Show Headers
Activate Windows
Go to Settings to activate Windows.
Type here to search
52°F
7:16 PM
3/22/2025

[redacted]

Action Required: Remote Support Installation for Northline Logistics
to: <ryanmarlucio@outlook.com> in 2 hours

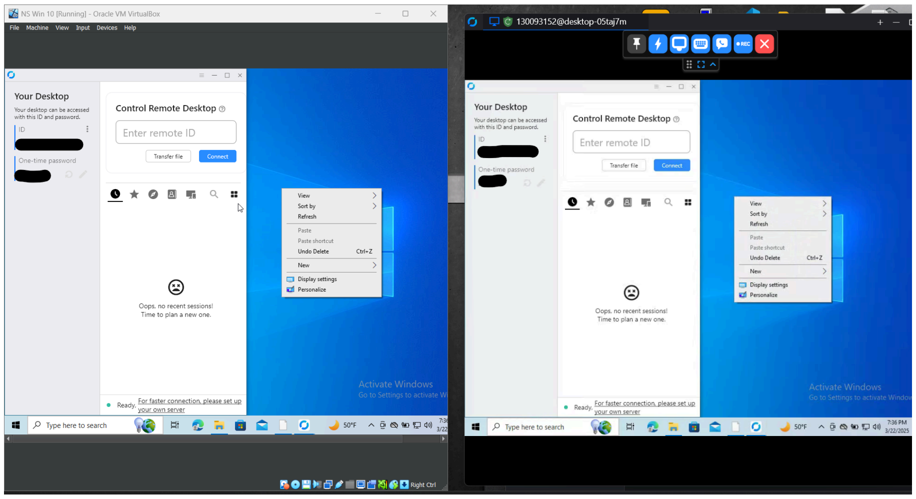
Action Required: Remote Support Installation...

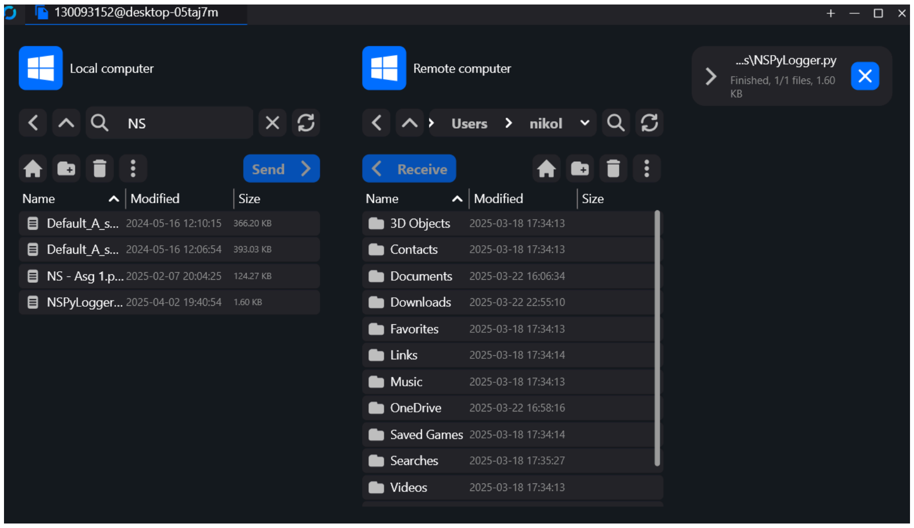
From: <it-support@northlinelogistics.com> 2025-03-23 00:54, 1.3 KB
To: Ryan Marlucio <ryanmarlucio@outlook.com>

```
Dear Northline Logistics Team Member,

As part of our ongoing efforts to enhance remote support for our hybrid
workforce, the Northline Logistics IT Department is implementing RustDesk as our
official remote assistance tool. This will allow us to troubleshoot technical
issues quickly and efficiently without Team Members needing to come into the
office.

Action Required:

Download and install RustDesk using the link below:
https://rustdesk.com/

After installation, please reply directly to this email with your RustDesk ID.

For your convenience, please use the “Reply” button instead of manually entering
an email address to ensure your response reaches the correct department.

If you experience any issues during installation, feel free to respond for
further assistance.

Thank you for your cooperation.

Best regards,
```

After running the phishing campaign, the victim (the VM) downloaded RustDesk and followed the setup instructions. From there, we, as the attacker, remotely connected to the victim using RustDesk—pretending to be IT support checking if everything was working correctly. Within that first remote session, we transferred our Python keylogger script- *NSPyLogger.py*[2] to the VM and had it run silently in the background.

[2] Our Python keylogger *NSPyLogger.py*, runs without opening the terminal, writes all keystrokes to a .txt file. Upon the press of the ESC key, the program terminates, and exports *log.txt* to the same directory that the keylogger is in. You can find the original program code in the .zip file accompanying our paper submission.


NS Win 10 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Your Desktop
Your desktop can be accessed with this ID and password.
ID
[redacted]
One-time password
[redacted]
Control Remote Desktop
Enter remote ID
Transfer file
Connect
Oops, no recent sessions!
Time to plan a new one.
Ready
For faster connection, please set up your own server
View
Sort by
Refresh
Paste
Paste shortcut
Undo Delete Ctrl+Z
New
Display settings
Personalize
Activate Windows
Go to Settings to activate Windows.
Type here to search
130093152@desktop-05taj7m


130093152@desktop-05taj7m

Local computer

NS

Send

Name	Modified	Size
Default_A_s...	2024-05-16 12:10:15	366.20 KB
Default_A_s...	2024-05-16 12:06:54	393.03 KB
NS - Asg 1.p...	2025-02-07 20:04:25	124.27 KB
NSPyLogger...	2025-04-02 19:40:54	1.60 KB

Remote computer

Users > nikol

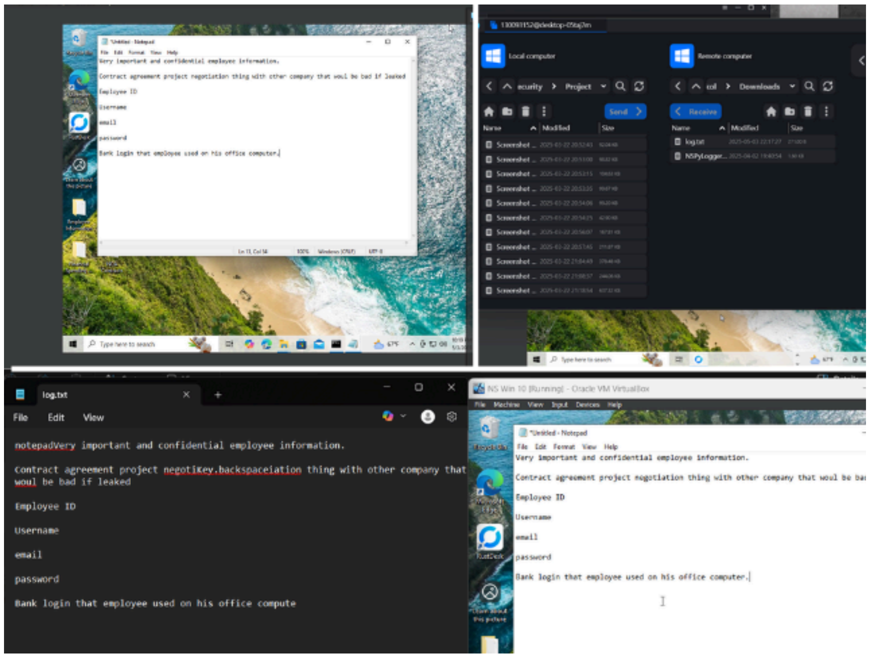
Receive

Name	Modified	Size
3D Objects	2025-03-18 17:34:13	
Contacts	2025-03-18 17:34:13	
Documents	2025-03-22 16:06:34	
Downloads	2025-03-22 22:55:10	
Favorites	2025-03-18 17:34:13	
Links	2025-03-18 17:34:14	
Music	2025-03-18 17:34:13	
OneDrive	2025-03-22 16:58:16	
Saved Games	2025-03-18 17:34:14	
Searches	2025-03-18 17:35:27	
Videos	2025-03-18 17:34:13	

...s\NSPyLogger.py
Finished, 1/1 files, 1.60 KB

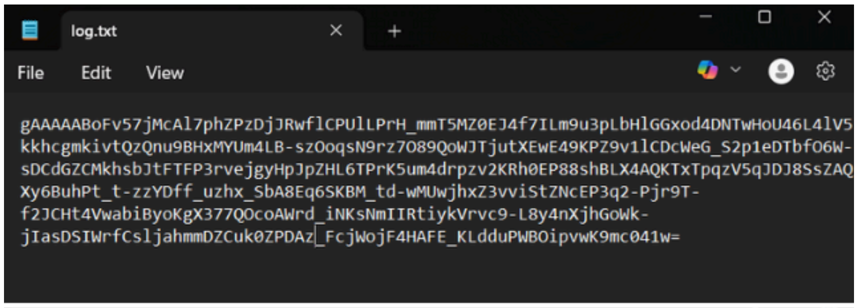
Later, we connected remotely again using RustDesk, and this time we retrieved the *log.txt* file outputted by the keylogger. And voilà—we had keystrokes. On top of that, we were able to access and make copies of simulated "confidential employee files" that hypothetically contained sensitive or private information. Then, we took those log files and encrypted them using a Python program we wrote *PyEncrypter.py*[3]. Finally, "we revealed ourselves" to the company and simulated a ransom demand.

[3] You can find the original program code (as well as our decrypter *PyDecrypter.py)* in the .zip file accompanying our paper submission.

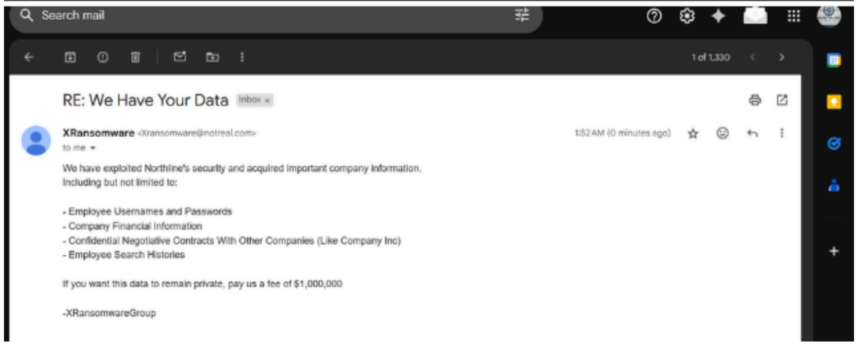


```
notepadVery important and confidential employee information.

Contract agreement project negotiKey.backspaceiation thing with other company that
woul be bad if leaked

Employee ID

Username

email

password

Bank login that employee used on his office compute
```

```
Very important and confidential employee information.

Contract agreement project negotiation thing with other company that woul be ba

Employee ID

Username

email

password

Bank login that employee used on his office computer.
```



gAAAAABoFv57jMcAl7phZPzDjJRwflCPUlLPrH_mmT5MZ0EJ4f7ILm9u3pLbHlGGxod4DNTwHoU46L4lV5
kkhcgmkivtQzQnu9BHxMYUm4LB-szOoqsN9rz7O89QoWJTjutXEwE49KPZ9v1lCDcWeG_S2p1eDTbfO6W-
sDCdGZCMkhsbJtFTFP3rvejgyHpJpZHL6TPrK5um4drpzv2KRh0EP88shBLX4AQKTxTpqzV5qJDJ8SsZAQ
Xy6BuhPt_t-zzYDff_uzhx_SbA8Eq6SKBM_td-wMUwjhxZ3vviStZNcEP3q2-Pjr9T-
f2JCHt4VwabiByoKgX377QOcoAWrd_iNKsNmIIRtiykVrvc9-L8y4nXjhGoWk-
jIasDSIWrfCsljahmmDZCuk0ZPDAz_FcjWojF4HAFE_KLdduPWBOipvwK9mc041w=



RE: We Have Your Data

XRansomware <Xransomware@notreal.com>
to me
1:52 AM (0 minutes ago)

We have exploited Northline's security and acquired important company information.
Including but not limited to:

- Employee Usernames and Passwords
- Company Financial Information
- Confidential Negotiative Contracts With Other Companies (Like Company Inc)
- Employee Search Histories

If you want this data to remain private, pay us a fee of $1,000,000

-XRansomwareGroup

We were able to complete every one of these steps despite the constraints. We decided to use Python instead of Batch for two main reasons: one, we're more comfortable in Python and it made the demonstration easier for us to pull off without sacrificing accuracy. And two, Python is a more dynamic programming language than Batch, so it still worked for our purposes.

Our results can be seen in the screenshots, but in short: we completed all steps successfully. Now we understand the dangers of phishing emails, and we also understand why it's so important for companies and their IT or Cybersecurity teams to not only respond to these attacks quickly, but also train their employees to recognize things like phishing attempts. Because let's be honest—basic awareness prevents most attacks.

To conclude this section: the most important takeaway—one that's honestly a little concerning—is just how *simple* (not necessarily easy, but simple) something like this was set up. It's easy to see how malicious groups with more experience and more resources could pull off something very real, very damaging, and very fast.

Conclusion

We believe that the work we did was good for the scope of this project. We tried to make it as close as possible to a real ransomware attack. But again, due to RATs being illegal, we settled for a phishing-to-remote-desktop setup as the closest thing we could do. However, this isn't too big of a compromise, because we still were able to demonstrate the keylogging portion of Interlock's ransomware attacks.

In regards to encrypting the "stolen" files. Usually, the files are encrypted on the victim's computer. We encrypted the files once they were on the host computer due to the aforementioned slowness of the VM. While this isn't necessarily accurate to what usually happens, it still accomplishes to demonstrate how it *could* be done; which *was* our main goal with this project. To demonstrate and outline the process of a ransomware attack, and despite some minute differences, I believe our demonstration was more than satisfactory.

Outside of that, other challenges we faced were: programming the keylogger and encrypter/decrypter scripts, the general slowness of the VM, and the email setup process. Regarding that last one—GoPhish itself was easy to use; the problem was setting up the victim email to properly receive the disguised phishing email. We tried everything: tinkering with SMTPs, and IMAPs. Tried different Email applications: Gmail, Outlook, ProtonMail, and Mailtrap.io. We ended up having to use Mailtrap.io, but that wasn't ideal because the user interface wasn't realistic; it didn't look or feel like an actual inbox. A realistic interface on Mailtrap.io would have required a premium subscription, which we were not willing to acquire.

If someone wants to take our work and build on it, we would recommend tackling this email realism issue. Also—we would encourage writing the programs in Batch instead of Python. Finally, we would suggest they encrypt the files on the "victim" machine. These would be good building blocks, and would make the demonstration even more realistic.

References

[1] Biasiotto, Elio, Johnson, Raghuprasad, and Szeliga, “Unwrapping the Emerging Interlock Ransomware Attack,” Cisco Talos Blog, 12 Dec. 2024. [Online]. Available: https://blog.talosintelligence.com/emerging-interlock-ransomware/

[2] 30, The Sunshine State May, et al., “‘Operation Endgame’ Hits Malware Delivery Platforms,” Krebs on Security, 30 May 2024. [Online]. Available: https://krebsonsecurity.com/2024/05/operation-endgame-hits-malware-delivery-platforms/

[3] Ojo, Abraham Olasunkanmi, “Ransomware Trends and Mitigation Strategies: A Comprehensive Review,” Global Journal of Engineering and Technology Advances, vol. 22, no. 3, pp. 9–16, 2025.

[4] Sudheer, Sooraj, “Ransomware Attacks and Their Evolving Strategies: A Systematic Review of Recent Incidents,” Journal of Technology and Systems, vol. 6, no. 7, pp. 32–59, 2024.

[5] Kim, Giyoon, et al., “How to Decrypt Files Encrypted by Rhysida Ransomware without the Attacker’s Private Key,” Computers & Security, vol. 151, 104340, 2025. [Online]. Available: https://doi.org/10.1016/j.cose.2025.104340

[6] O’Kane, Patrick, et al., “Evolution of Ransomware,” IET Networks, vol. 7, no. 6, pp. 373–379, 2018. [Online]. Available: https://doi.org/10.1049/iet-net.2018.5020