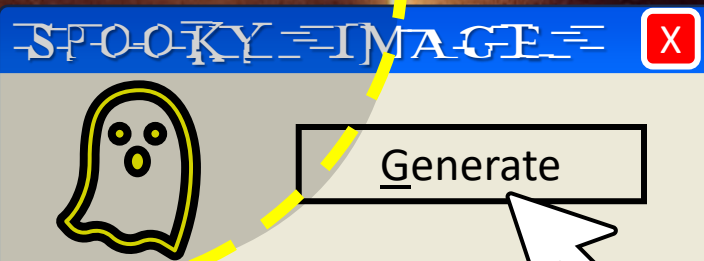


OCTOBER 2023



✕ HACKS OF THE MONTH	03
✕ CYBER NEWS UPDATES	13
✕ JOBS AND INTERNSHIPS	20
✕ QUICK PROJECT	22
✕ ADVANCED PROJECT	31
✕ FACULTY CORNER	36



# WE WANT YOU FOR THE SOC

Join the UofA's Security Operations Center today for a rewarding, for credit, cybersecurity intern experience!

## What Will You Do As A SOC Intern?

### INVESTIGATION



Review vulnerability data, and record and track IT security incidents, including:

- Compromised Accounts
- Phishing
- Abuse reports

#### CORE OF THE INTERNSHIP

### OPERATIONS



Get hands-on experience with security tools and practices within a professional business environment:

- SIEM
- IPS
- Netflow

#### EXPERIENCE THE SOC WORKFLOW

### HUNTING



Perform threat hunting to detect and eradicate threats using various paid and open source intelligence tools!

#### LEARN AND USE OSINT SKILLS

This internship is available to be taken **for credit** with advisor approval and provides opportunities to develop your skills as a professional in the industry.

Interested? Apply Now On Handshake:

<https://app.joinhandshake.com/emp/jobs/8260025>



## MINIMUM Qualifications



## PREFERRED Experience

- Located in **Tucson, Arizona**
- Access to **reliable internet connection** and **computing resources**
- Internship is available for **credit** — with **advisor approval**
- **15-25 hours** per week **Mon-Fri || 9a -> 5pm**
- Must be a **current UofA student** studying **Cyber Operations, Computer Science, or related degree**
- The **Incident Handling Process**
- **Networking** (TCP/IP, UDP, DNS, DHCP, HTTP, etc.)
- **Security technologies and concepts** (Firewalls, Network Intrusion Detection systems, SIEM, CIA Triad)
- **NIST Cybersecurity Framework**
- **Common data analysis** tools and techniques
- Understanding of **Information Security best practices** at a **individual and/or organizational level**

Questions or concerns?  
Email: [security@arizona.edu](mailto:security@arizona.edu)

# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

In cybersecurity, threat actors continually adapt, employing sophisticated tools and techniques to breach defenses and exploit vulnerabilities. Among the emerging threats, a group known as ShadowSyndicate has recently been spotlighted for its extensive ransomware operations and affiliation with various ransomware-as-a-service (RaaS) campaigns. This group, previously identified as Infra Storm, has been linked to a series of cyber-attacks deploying multiple ransomware families across a network of 85 servers, showcasing a new level of coordination and threat.

ShadowSyndicate's activities were first observed in July 2022, with its malicious footprint extending well into 2023. The group's modus operandi involves deploying a range of ransomware, including Quantum, Nokoyawa, BlackCat/ALPHV, Clop, Royal, Cactus, and Play ransomware, indicating a broad spectrum of cyber-attack capabilities. Identifying a distinct SSH fingerprint across the servers used by ShadowSyndicate has been a crucial lead for the researchers, aiding in tracing the group's malicious activities.

Moreover, the cyber threat landscape is witnessing a shift in the tools employed by hackers. The Sliver toolkit, for instance, is gaining traction among threat actors as an alternative to the Cobalt Strike penetration testing suite, which has been a popular choice among hackers for its ability to drop "beacons" on compromised networks, facilitating lateral movement to high-value systems. The adoption of Sliver, known for its capability to evade Endpoint Detection and Response (EDR) and antivirus solutions, underscores the continual adaptation of threat actors to bypass evolving security measures.



# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

The collaborative efforts of security researchers have been instrumental in unveiling the extensive network and operations of ShadowSyndicate. Their findings show the group's affiliation with multiple ransomware operations, possibly acting as an initial access broker (IAB) or an affiliate to various ransomware campaigns. The investigation also highlights using Cobalt Strike command and control machines, a hallmark of ShadowSyndicate's operations.

The analysis of ShadowSyndicate's activities and the rising adoption of alternative hacking tools like Sliver paints a concerning picture of the evolving threat landscape. It underscores the imperative for continuous advancements in cybersecurity measures, collaborative research, and information sharing among the global cybersecurity community to stay ahead of malicious actors and ensure a robust defense against the burgeoning threat of cyber-attacks.

## DUSTING FOR FINGERPRINTS: SHADOWSYNDICATE, A NEW RAAS PLAYER?

The article discusses the identification and activities of a cyber threat group named ShadowSyndicate, which has been linked to multiple ransomware operations across 85 servers. Security researchers have traced this group's deployment of seven ransomware families over the past year. The researchers believe that ShadowSyndicate could be an initial access broker (IAB) or an affiliate to various ransomware campaigns based on the distinct SSH fingerprint found on the servers used by the group.

The article also mentions using Cobalt Strike command and control machines in ShadowSyndicate's operations, indicating sophistication in their cyber-attack capabilities.



# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

The emergence of ShadowSyndicate as a significant player in the cyber threat landscape highlights the evolving nature of cyber threats and the sophistication of modern-day hackers. The group's ability to deploy multiple ransomware families across a vast network of servers showcases a high level of coordination and technical expertise. Identifying a unique SSH fingerprint across these servers provides a crucial lead for researchers to trace and possibly counter the group's malicious activities.

The collaborative efforts between various cybersecurity entities, including Group-IB, Bridewell, and independent researchers, underline the importance of collective action in combating cyber threats. Sharing intelligence and resources is vital in understanding the modus operandi of threat groups like ShadowSyndicate and devising effective strategies to mitigate their risks.

Furthermore, the use of Cobalt Strike command and control machines by ShadowSyndicate indicates a preference for sophisticated tools that can potentially bypass conventional security measures. This calls for continuously evolving cybersecurity strategies to stay ahead of the curve in identifying and neutralizing threats from well-organized cyber-attack groups.


The article also hints at the possibility of ShadowSyndicate acting as an initial access broker or an affiliate to various ransomware campaigns, which, if confirmed, could reveal a broader network of cyber criminal alliances.

# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

Such alliances could significantly amplify the threat posed by individual groups, making securing cyber infrastructures increasingly challenging.

In conclusion, the activities of ShadowSyndicate underscore the need for enhanced cybersecurity measures, collaborative research, and a proactive approach to anticipating and countering emerging cyber threats.

## SHADOWSYNDICATE HACKERS LINKED TO MULTIPLE RANSOMWARE OPS, 85 SERVERS

The article delves into the investigative findings concerning a cyber threat actor known as ShadowSyndicate, believed to have deployed seven different ransomware families in various attacks over the past year. The collaborative investigation has identified a distinct SSH fingerprint across 85 IP servers, linking them to ShadowSyndicate's malicious activities. The servers, mostly tagged as Cobalt Strike command and control machines, have been active since July 16, 2022, and were still in use as of August 2023. The researchers speculate that ShadowSyndicate could be an initial access broker (IAB) or an affiliate to multiple ransomware operations. The article also mentions a transfer of at least 12 IP addresses from notorious ransomware operators to ShadowSyndicate, hinting at a possible connection, although a high-confidence link remains elusive.

# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

The detailed investigation into ShadowSyndicate's operations reveals the intricate and evolving nature of cyber threat actors. Using a distinct SSH fingerprint across a vast network of servers demonstrates sophistication and operational security that pose significant challenges to cybersecurity efforts. The potential affiliation of ShadowSyndicate with various ransomware operations suggests a collaborative or hierarchical structure among cybercriminal groups, which could enhance their capabilities and resilience against countermeasures.

The utilization of Cobalt Strike command and control machines by ShadowSyndicate indicates a preference for advanced tools that can potentially evade detection and facilitate large-scale cyber-attacks. This highlights the need for continuous advancements in cybersecurity technologies and methodologies to combat such well-equipped threat actors effectively.

The potential connection between ShadowSyndicate and other notorious ransomware operators, as indicated by the transfer of IP addresses, suggests a fluid and possibly collaborative cybercriminal ecosystem. This could lead to the sharing of resources, tactics, and intelligence among different threat actors, thereby amplifying the risks they pose to organizations and individuals alike.

Furthermore, the open invitation by Group-IB for external researchers to collaborate in uncovering the remaining obscure parts of ShadowSyndicate's operations underscores the importance of collective intelligence and collaborative efforts in the fight against cybercrime. Such collaborative endeavors could significantly enhance the understanding of threat actors like ShadowSyndicate and contribute to developing more effective countermeasures.

# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

In conclusion, the findings on ShadowSyndicate's activities are a stark reminder of the complex and evolving threat landscape. The collaborative approach adopted by researchers in this investigation sets a positive precedent for future endeavors aimed at unmasking and neutralizing cyber threat actors.

## TRANSITION TO STEALTH: HACKERS' ADOPTION OF SLIVER TOOLKIT

The article discusses the shift of threat actors from the widely used Cobalt Strike penetration testing suite to lesser-known frameworks like Sliver to evade detection. This transition is driven by the enhanced ability of defenders to detect and stop attacks relying on Cobalt Strike. Sliver, an open-source, cross-platform kit, is becoming a favored alternative due to its capabilities to bypass Endpoint Detection and Response (EDR) and antivirus solutions. The article mentions that cybercrime gangs and state-sponsored groups have been observed utilizing Sliver in their intrusion campaigns. Microsoft has provided a set of tactics, techniques, and procedures (TTPs) to help defenders identify and counter malicious activities facilitated by Sliver and other emerging Command and Control (C2) frameworks.

The migration from Cobalt Strike to Sliver and other lesser-known frameworks underscores the adaptive nature of cyber threat actors. As defensive measures evolve, so do the tactics and tools employed by hackers. This cat-and-mouse dynamic necessitates continual advancements in cybersecurity methodologies to stay ahead or at least on par with adversarial tactics.





# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

The adoption of Sliver, as noted in the article, is a testament to the ongoing search by threat actors for more stealthy and effective means of infiltrating networks. The toolkit's ability to evade common detection mechanisms presents a significant challenge to defenders, emphasizing the need for continuous research and development in cybersecurity.

The article also highlights the diverse range of threat actors, from cybercrime gangs to state-sponsored groups, adopting Sliver for their operations. This broad adoption spectrum suggests that the toolkit is effective and appealing to different actors, regardless of their scale or objectives.

Microsoft's proactive approach in providing detection guidance for Sliver-based activities is a positive step towards empowering defenders. By sharing knowledge on identifying and counter threats posed by Sliver and similar frameworks, the cybersecurity community can better prepare for and respond to evolving threats.

Furthermore, the article underscores the importance of threat intelligence sharing and collaborative defense strategies. As threat actors continually adapt their tactics, a collective effort among cybersecurity stakeholders is crucial to combat and mitigate the risks posed by emerging threats effectively.

In conclusion, the shift towards Sliver and similar frameworks indicates the evolving threat landscape. The proactive measures by organizations like Microsoft and collaborative efforts within the cybersecurity community are vital in developing robust defense mechanisms to counter these evolving threats.

# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

## CONCLUSION

The three articles collectively paint a picture of an evolving cyber threat landscape characterized by the emergence of sophisticated threat actors and a shift towards stealthier attack frameworks. Here are the key takeaways from the analysis of these articles:

### Emergence of Sophisticated Threat Actors:

- ShadowSyndicate emerges as a notable threat actor, orchestrating multiple ransomware campaigns and leveraging a variety of ransomware families. Their operations, spanning over a year, demonstrate high sophistication and adaptability.
- The group's affiliation with various ransomware operations and its potential role as an initial access broker (IAB) underline modern cyber-criminal enterprises' complex and organized nature.

### Shift to Stealthier Attack Frameworks:

- Threat actors are transitioning from well-known frameworks like Cobalt Strike to lesser-known yet effective alternatives like Sliver to evade detection.
- The adoption of Sliver, a toolkit initially designed for security testing by cybercrime gangs and state-sponsored actors, underscores the toolkit's effectiveness in bypassing contemporary defensive measures.

# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

## Collaborative Defense and Threat Intelligence Sharing:

- The collaborative efforts between different cybersecurity firms and researchers in identifying and analyzing the activities of ShadowSyndicate and the adoption of Sliver reflect a proactive approach toward understanding and mitigating threats.
- Microsoft's initiative in providing detection guidance for Sliver-based activities exemplifies the importance of threat intelligence sharing and community-driven defense strategies.

## Continuous Evolution of Defensive Measures:

- As threat actors adapt their tactics and tools, the necessity for continuous evolution in defensive measures is highlighted. The development of new detection and response strategies, along with the adoption of proactive threat-hunting practices, is crucial to stay ahead of adversaries.

## Challenges in Attribution and Detection:

- The articles also touch on the challenges faced in attributing malicious activities to specific threat actors and detecting stealthier malware operations, emphasizing the need for advanced analytical tools and methodologies.

# Unmasking ShadowSyndicate: A Deep Dive into Emerging Cyber Threat Landscape

## Need for Broader Cybersecurity Community Engagement:

- The invitation for external researchers to collaborate in uncovering the obscure parts of ShadowSyndicate's operations and Microsoft's sharing of detection guidance underscore the importance of broader engagement within the cybersecurity community to combat evolving threats effectively.

In summary, the evolving tactics of threat actors like ShadowSyndicate and the shift towards stealthier attack frameworks like Sliver underline the dynamic and complex nature of the modern cyber threat landscape. The collaborative efforts among cybersecurity stakeholders and the continuous evolution of defensive measures are imperative to effectively address and mitigate the risks posed by these emerging threats.



# LuaDream's Lullaby: A Tale of Sandman's Stealthy Telecom Infiltrations

Recently, the cyber realm has witnessed the emergence of a clandestine threat actor dubbed 'Sandman,' whose activities have sent ripples through the telecommunications sector across the Middle East, Western Europe, and South Asia. The elusive nature of Sandman, coupled with its sophisticated malware arsenal, presents a quintessential case of modern cyber-espionage campaigns aimed at pilfering sensitive information from high-value targets.

Sandman's modus operandi is emblematic of a well-orchestrated attempt to maintain a low profile while navigating through the networks of telecommunication service providers. The initial ingress is often facilitated through stolen administrative credentials, paving the way for a series of "pass-the-hash" attacks. These attacks exploit the NTLM hashes stored in memory to authenticate to remote servers and services. This tactic underscores the actor's adeptness at leveraging system vulnerabilities for lateral movement within compromised networks.

At the heart of Sandman's technical prowess is a modular info-stealing malware named 'LuaDream,' a nefarious creation designed to extend its tentacles deep into the infected systems, exfiltrating valuable data while managing a suite of plugins to augment its malicious functionalities. LuaDream's architecture is a testament to Sandman's meticulous approach to evading detection. The malware employs a seven-step in-memory staging process initiated by exploiting the Windows Fax or Spooler service to load its malicious payload. This intricate staging process, laden with anti-analysis measures such as concealing threads from debuggers and employing XOR-based encryption, epitomizes the lengths to which Sandman goes to cloak its activities.

The LuaDream malware isn't just a standalone menace; it's a cog in a larger machine orchestrated by Sandman. Once nestled within the compromised systems, LuaDream establishes a covert communication channel with its command and control (C2) server, transmitting gathered intelligence while awaiting further instructions. The malware's modular nature allows for the deployment of specific plugins, each tailored for different nefarious purposes, showcasing Sandman's ability to adapt and evolve in response to the defensive measures employed by the targeted entities.

The telecommunications sector's allure for threat actors like Sandman isn't fortuitous. These entities are repositories of a vast swath of sensitive data, making them prime targets for state-sponsored actors and mercenary groups. The recent surge in cyber-espionage campaigns against telecom providers underscores a broader narrative of an escalating cyber arms race, where the stakes are perpetually rising.

As Sandman continues to elude attribution, its activities are a stark reminder of the evolving threat landscape. The tale of Sandman and LuaDream is but a glimpse into the murky waters of cyber espionage, offering a narrative replete with lessons for aspiring cybersecurity aficionados.

# LuaDream's Lullaby: A Tale of Sandman's Stealthy Telecom Infiltrations

## SANDMAN APT: A MYSTERY GROUP TARGETING TELCOS WITH A LUAJIT TOOLKIT

The article unveils a new threat actor, Sandman APT, identified by SentinelLabs in collaboration with QGroup GmbH, targeting telecommunication providers primarily in the Middle East, Western Europe, and the South Asian subcontinent. The threat actor's activities are characterized by strategic lateral movements and minimal engagements to minimize detection risk. Sandman APT has deployed a novel modular backdoor malware named LuaDream, utilizing the LuaJIT platform, a relatively rare occurrence in the threat landscape.

The LuaDream malware is a well-structured, actively developed project with a modular architecture allowing for the management of attacker-provided plugins and exfiltration of system and user information. The malware's staging process is designed to evade detection and thwart analysis while deploying the malware directly into memory. LuaDream's implementation leverages the LuaJIT platform to make malicious Lua script code difficult to detect.

The article also hints at the espionage motivations behind these activities, given the sensitive data telecommunication providers hold. The geographical distribution of victims and the malware's development efforts dating back to the first half of 2022 are also highlighted. The article suggests the possibility of a private contractor or mercenary group being behind Sandman APT due to certain inconsistencies observed in their operational practices.

### Technical Sophistication:

- The article underscores the technical sophistication of Sandman APT, particularly through the deployment of the LuaDream malware. Using LuaJIT, a platform not commonly associated with APT malware, alongside a modular architecture, indicates high technical capability and adaptability.

### Espionage Motivations:

- The targeting of telecommunication providers, known for holding sensitive data, alongside the strategic lateral movements within compromised networks, strongly suggests espionage motivations. This aligns with a broader trend of state-sponsored or mercenary groups targeting critical infrastructure for intelligence gathering.

### Attribution Challenges:

- The article highlights the challenges in attributing the Sandman APT to any known threat actors. The inconsistencies between the high-end development of the malware and poor segmentation practices lead to the possibility of a private contractor or mercenary group, showcasing the complexities in attribution in modern cyber-espionage campaigns.



# LuaDream's Lullaby: A Tale of Sandman's Stealthy Telecom Infiltrations

## Evasion Techniques:

- The detailed evasion techniques employed by LuaDream, including in-memory execution and anti-analysis measures, reflect a growing trend among threat actors to develop malware capable of evading detection and analysis, thereby prolonging their presence within compromised environments.

## Educational Value:

- For cybersecurity students, this article serves as a rich source of information on how modern-day threat actors operate, the technical intricacies of advanced malware, and the challenges faced by the cybersecurity community in attributing and mitigating sophisticated threats.

The Sandman APT's activities, as detailed in the article, provide a glimpse into the evolving and complex landscape of cyber espionage, underscoring the necessity for continuous advancements in cybersecurity practices to counter such threats.

## HACKERS BACKDOOR TELECOM PROVIDERS WITH NEW HTTPSNOOP MALWARE

The article discusses a cyber threat orchestrated by the 'Sandman' group, targeting telecommunication service providers in the Middle East. This threat is characterized by deploying two novel malware strains, HTTPSnoop and PipeSnoop. According to a report by Cisco Talos, these malware implants, part of an intrusion set named 'ShroudedSnooper,' serve different operational goals concerning the level of infiltration. Both implants masquerade as security components of the Palo Alto Networks Cortex XDR product to evade detection, a tactic that echoes the sophisticated evasion techniques observed in Sandman's LuaDream malware, as discussed in Article 1.

HTTPSnoop monitors HTTP(S) traffic on an infected device for specific URLs, decoding incoming base64-encoded data from these URLs and executing it as a shellcode on the compromised host. It activates on the target system via DLL hijacking, a technique also employed by Sandman's LuaDream malware, and sets up a backdoor web server to process incoming HTTP requests. Three variants of HTTPSnoop have been identified, each with different URL listening patterns to mimic legitimate traffic, making malicious requests nearly indistinguishable from benign traffic.

On the other hand, PipeSnoop, spotted in May 2023, acts as a backdoor that executes shellcode payloads on breached endpoints through Windows IPC (Inter-Process Communication) pipes. Unlike HTTPSnoop, which targets public-facing servers, PipeSnoop is suited for operations deep within compromised networks, reflecting a similar depth of infiltration as observed with Sandman's LuaDream malware.



# LuaDream's Lullaby: A Tale of Sandman's Stealthy Telecom Infiltrations

The article emphasizes telecom service providers' critical role in managing sensitive information and critical infrastructure, making them attractive targets for state-sponsored threat actors like Sandman. The surge in attacks against telecom entities underscores the urgent need for enhanced security measures and international cooperation.

## Malware Sophistication:

- The article, when read in conjunction with the details from Article 1, highlights Sandman's technical sophistication and diversified malware arsenal. The operational capabilities of HTTPSnoop and PipeSnoop, along with the previously discussed LuaDream malware, demonstrate high technical expertise and adaptability.

## Operational Diversity:

- The distinct operational goals served by HTTPSnoop, PipeSnoop, and LuaDream indicate Sandman's well-organized and diversified attack strategy. This diversity allows the threat actors to infiltrate different layers of the targeted networks, showcasing a multi-faceted approach to achieving their objectives.

## Evasion Techniques:

- The evasion techniques employed by both HTTPSnoop and PipeSnoop, such as mimicking legitimate URL patterns and masquerading as genuine security components, underline the growing challenge of detecting and mitigating such threats. These techniques also reflect a broader trend of threat actors like Sandman employing increasingly sophisticated evasion and obfuscation methods to avoid detection.

## Target Selection:

- The targeting of telecommunication service providers, a critical sector, reiterates the strategic importance of such entities for state-sponsored threat actors like Sandman. The sensitive data and critical infrastructure these providers manage make them high-value targets for espionage and sabotage activities.

## Educational Implications:

- For cybersecurity students, this article, along with the insights from Article 1, provides a comprehensive view of the evolving threat landscape, the technical intricacies of modern malware, and the importance of robust cybersecurity measures in protecting critical infrastructure. It also serves as a case study on how threat actors like Sandman diversify their tactics to achieve their objectives.

The detailed analysis of HTTPSnoop and PipeSnoop within the article, along with the insights on Sandman's LuaDream malware from Article 1, sheds light on the continuous evolution of malware and the escalating threat to critical sectors like telecommunications, emphasizing the need for vigilance and advanced security measures to counter such threats.



# LuaDream's Lullaby: A Tale of Sandman's Stealthy Telecom Infiltrations

## 'Sandman' Hackers Backdoor Telcos with New LuaDream Malware

The article unveils a cyber espionage campaign orchestrated by a previously unknown threat actor, Sandman, targeting telecommunication service providers across the Middle East, Western Europe, and South Asia. This campaign is part of a larger narrative where Sandman has been associated with sophisticated cyber-attacks on telecommunication sectors, as discussed in the first article. The malware employed in this campaign, named 'LuaDream,' was discovered by SentinelLabs in collaboration with QGroup GmbH in August 2023. Sandman's operational style is characterized by a low-profile approach, strategic lateral movement within compromised networks, and long-term access to breached systems, aiming to maximize its cyber espionage operations.

Upon breaching a network using stolen administrative credentials, Sandman utilizes "pass-the-hash" attacks to authenticate to remote servers and services by extracting and reusing NTLM hashes stored in memory. The targeted workstations, particularly those assigned to managerial personnel, indicate Sandman's interest in privileged or confidential information.

LuaDream malware, deployed using DLL hijacking, is a sophisticated tool designed for data collection and managing plugins that extend its functionality. The malware's development appears to be active, with a version string indicating ongoing updates and logs and testing functions dating back to June 2022. LuaDream's staging process is meticulously designed to evade detection, employing a seven-step in-memory process initiated by either the Windows Fax or Spooler service.

The article underscores the growing trend of advanced threat actors like Sandman targeting telecom companies for espionage, given the sensitive nature of the data they manage. It also references a similar recent activity, 'ShroudedSnooper,' discussed in the second article, that employed two novel backdoors, HTTPSnoop and PipeSnoop, against telecommunication carriers in the Middle East.

### Technical Sophistication:

- Sandman's LuaDream malware exemplifies high technical sophistication, showcasing a modular design, strategic evasion techniques, and a well-thought-out deployment strategy. The use of DLL hijacking and a complex in-memory staging process highlights the advanced technical capabilities of Sandman, aligning with the technical prowess demonstrated in the first article.

### Operational Stealth:

- Sandman's low-profile operational style, aimed at evading detection while achieving its espionage objectives, reflects a mature and calculated approach to cyber espionage. This stealthy approach is crucial for maintaining long-term access to sensitive targets.



# LuaDream's Lullaby: A Tale of Sandman's Stealthy Telecom Infiltrations

## Target Selection:

- The targeting of telecommunication service providers, a sector crucial for economic and national security, underscores the strategic objectives behind Sandman's campaign. The focus on managerial personnel suggests an interest in high-value, possibly strategic information.

## Malware Evolution:

- The active development of LuaDream, as indicated by the version string and the presence of logs and testing functions, suggests that Sandman is continually refining its tools to adapt to new challenges and maximize its espionage capabilities.

## Educational Implications:

- For cybersecurity students, this article provides a rich case study on modern cyber espionage campaigns. It offers insights into advanced malware design, operational tactics of threat actors, and the strategic significance of target selection in cyber espionage.

The detailed exposition of Sandman's LuaDream malware and its operational tactics in the article provides a comprehensive view of the threat posed by advanced cyber espionage campaigns to critical sectors like telecommunications. It emphasizes the importance of robust cybersecurity measures and international cooperation in countering such sophisticated threats, drawing a clear line of sophisticated cyber-attacks from Sandman, as seen in the previous articles.

The trio of articles meticulously unravels the clandestine operations of the elusive threat actor, Sandman, whose cyber espionage campaigns have been meticulously orchestrated against telecommunication service providers across the Middle East, Western Europe, and South Asia. Through a blend of sophisticated malware tools - LuaDream, HTTPSnoop, and PipeSnoop - Sandman has demonstrated high technical sophistication and operational stealth, hallmarks of an advanced persistent threat actor. The intricate narrative of Sandman's cyber espionage campaigns serves as a stark reminder of the dynamic and adversarial nature of the cybersecurity domain. The saga of Sandman serves as both a cautionary tale and a call to action, urging aspiring cybersecurity professionals to equip themselves with the knowledge, skills, and ethical grounding necessary to navigate the complex and ever-evolving cybersecurity landscape.



WiCyS – UArizona Student Chapter Presents:

# **RESUME BUILDING & COVER LETTER WORKSHOP**

The University of Arizona Writing Center

**Oct 2<sup>nd</sup> - 6:00 p.m.**

**Zoom**

Free and Open to the Public

<https://arizona.zoom.us/j/85472545975?pwd=R2R3eXpJRUNrbFk2WTRtS2x0S2VnZz09>

# GG13-2210 IT SPECIALIST (SYSANALYSIS) – CSAT Cyber Analyst

## Specialized Experience:

Experience in Operational cyber operations (Red Teams, Cyber Security Service Providers, Cyber Protective Teams, Cyber Forensics, Cyber Intelligence, etc.); DoD Information Technology/National Security Systems (IT/NSS) cybersecurity and cyber survivability test and evaluation planning, execution, analysis, and reporting; use of automated and manual cybersecurity and vulnerability assessment tools and techniques; coordination and conduct of penetration tests and red team assessments; and developing and presenting test concepts, plans, reports, and briefings to senior leaders.

## Responsibilities

- Implement applicable governmental and DoD acquisition and Cybersecurity (CS) policies and directives in planning, conducting, and reporting on the CS Testing and Evaluation of IT/NSS.
- Develop test strategies and approaches for evaluating program functional, operational, interoperability, and CS requirements.
- Identify, coordinate, and manage resources, schedules, and budgets supporting effective and efficient program CS Testing and Evaluation conduct.
- Identify, analyze, and propose methods for mitigating threats, vulnerabilities, and risks identified through CS Testing and Evaluation.
- Develop organizational CS Testing and Evaluation policies, processes, and support capabilities.

## Place of Employment

- Defense Systems Agency (DISA), Joint Interoperability Test Command (JITC) (<https://jitc.fhu.disa.mil/>), Fort Huachuca, AZ

Resumes for recent or soon-to-graduate UA students with training, skills, and experience supporting these areas would be greatly appreciated!

If anyone has any questions about this position, please contact **Chief Powell** for additional information.

DISA





WiCyS UArizona Student Chapter Presents:

# 2023 Cybersecurity Speaker Series

## Amy Justice

Sr. Manager, IT Security, Risk, & Compliance, Randstad USA

**Oct 16th - 6:00 p.m.**

Zoom

Open to the public.

<https://arizona.zoom.us/j/85472545975?pwd=R2R3eXpJRUNrbFk2WTRtS2x0S2VnZz09>

# HackO'Lantern: Python based scareware

Welcome to the eerie realm of HackO'Lantern, a spooky project designed to give cybersecurity enthusiasts a harmless scare. This program simulates a malware infection on your system, creating a ghostly atmosphere perfect for Halloween or any cybersecurity demonstration. However, fear not! Despite its sinister appearance, HackO'Lantern is harmless and will not cause any real damage to your system.

## DISCLAIMER:


Before diving into the abyss, it's crucial to understand that HackO'Lantern is meant solely for **educational and entertainment purposes**. Distributing this program without explaining its harmless nature could cause panic and concern. So, **share responsibly!**

In this walkthrough, we'll dissect HackO'Lantern, exploring the dark corners of its code to understand how it conjures its spooky effects. This hands-on experience is a fantastic way to learn by practicing, and by the end, you'll have a ghostly program to spook your friends.

## A WORD OF CAUTION:

While HackO'Lantern is harmless, altering its code to perform malicious actions could land you serious trouble, including **criminal charges** or expulsion. So, keep the frights friendly, and remember, it's all in good fun!

As we traverse through the haunted halls of HackO'Lantern's code, you'll learn how each eerie effect is achieved. For the grand finale, we'll package HackO'Lantern into a binary executable that can haunt Windows systems. To add a final touch of terror, we'll include a spooky icon to make the project fun and inviting.

Name	Date modified	Type	Size
 HackO'Lantern	9/20/2023 12:26 PM	Application	11,104 KB

We encourage you to tinker with HackO'Lantern, tweaking its code to create personalized scareware. However, remember to keep your creations harmless and share them responsibly. Now, let's descend into the spooky spectacle that is HackO'Lantern, and may your journey through its code be both enlightening and eerie!

Before we embark on this eerie expedition, check out the complete project and binary on its GitHub repository: [HackO'Lantern on GitHub](#).

## IMPORTING THE NECESSARY LIBRARIES:

The first step in our spooky journey involves importing the necessary libraries to ensure our program runs smoothly. Here's a breakdown of the imports and why they are crucial for HackO'Lantern:

- **tkinter** (as tk): This library is essential for creating our program's graphical user interface (GUI) elements, such as dialog boxes.
- **os**: The os module provides a way of using operating system-dependent functionality, like reading or writing to the file system.
- **string**: This module helps process standard Python strings and is used here to generate random strings.
- **winsound**: This module provides access to the basic sound-playing machinery provided by Windows platforms, enabling us to play spooky sounds.
- **random**: The random module generates random numbers essential for creating unpredictable, spooky effects.

```
1 import tkinter as tk
2 import os
3 import string
4 import winsound
5 import random
```

## ENCAPSULATING THE SPOOKINESS IN SPOOKYAPP CLASS:

Encapsulating all the spooky functionalities within a class called **SpookyApp** makes our code organized, reusable, and easy to manage. It follows the Object-Oriented Programming (OOP) paradigm, a good practice in software development.

## Function Breakdown:

Now, let's dissect each function within the SpookyApp class to understand the magic behind the spookiness.

- **init(self):** This constructor method initializes the Tkinter window and binds the F9 key to exit the application.

```
7  class SpookyApp:
8  def __init__(self):
9      self.root = tk.Tk()
10     self.root.withdraw() # Hide the main window
11     self.dialog_count = 0
12     self.root.bind('<F9>', self.exit_app) # Bind F9 key to exit function
```

- **create\_string(self):** Generates a random string, which could be used for various spooky effects.

```
14     def create_string(self):
15         return "".join([random.choice(string.ascii_letters) for x in range(random.randint(5, 20))])
16
```

- **spooky\_beeps(self):** Plays a series of random beeps to create a spooky ambiance. I decided to disable this as it was too much during testing.

```
17     def spooky_beeps(self):
18         for _ in range(10): # Play 10 beeps
19             freq = random.randint(200, 2000) # Random frequency between 200 and 2000 Hz
20             duration = random.randint(100, 500) # Random duration between 100 and 500 ms
21             winsound.Beep(freq, duration)
```

- **spooky\_message(self):** Returns a random spooky message to be displayed in the dialog boxes.

```
23     def spooky_message(self):
24         messages = [
25             "I see you... \U0001F47B", # Ghost emoji
26             "Why did you run me? \U0001F480", # Skull emoji
27             "Do you believe in ghosts? \U0001F47B",
28             "You can't escape... \U0001F63F", # Crying cat face emoji
29             "Happy Halloween! \U0001F383" # Jack-o-lantern emoji
30         ]
31         return random.choice(messages)
```



- **fake\_delete\_files(self):** Simulates the deletion of files by printing a message to the console, adding to the scare factor.

```
33 def fake_delete_files(self):
34     base_directory = os.path.expanduser('~') # Get the user's home directory
35
36     # List of common file extensions for documents, images, etc.
37     important_extensions = ['.doc', '.docx', '.pdf', '.jpg', '.jpeg', '.png', '.txt', '.xls', '.xlsx', '.ppt', '.pptx']
38
39     # List of folders to search
40     target_folders = ['Documents', 'Pictures', 'Desktop', 'Downloads']
41
42     # Recursively search for files with the specified extensions in the target folders
43     files = []
44     for folder in target_folders:
45         folder_path = os.path.join(base_directory, folder)
46         if os.path.exists(folder_path): # Check if the folder exists
47             for dirpath, dirnames, filenames in os.walk(folder_path):
48                 for filename in filenames:
49                     if any(filename.endswith(ext) for ext in important_extensions):
50                         files.append(os.path.join(dirpath, filename))
51
52     if files:
53         selected_file = random.choice(files)
54         print(f"Deleting the following file because it is not scary: {selected_file} ...")
55     else:
56         print("No important-looking files found to delete!")
57
58     self.root.after(2000, self.fake_delete_files) # Schedule the next call
```

In the **fake\_delete\_files(self):** function, we delve into a simulated realm of file deletion, a common scare tactic employed by real-world malware to instill fear and urgency in its victims. By targeting specific file extensions such as **.doc**, **.docx**, **.pdf**, **.jpg**, and others, we mimic the behavior of malicious software aiming to erase valuable data. These extensions represent common document, image, and presentation formats that most users would dread losing.

The function's eerie journey begins in the user's home directory, specifically targeting folders like 'Documents', 'Pictures', 'Desktop', and 'Downloads.' These folders are often the repositories of personal and important files, making them the perfect targets for our simulated scareware. By recursively searching through these folders, HackO'Lantern creates an illusion of scanning the system for files to delete, further enhancing the spooky malicious experience.

The **fake\_delete\_files(self):** function encapsulates the essence of scareware, providing a safe yet spooky simulation of malware operation. It's a ghostly reminder of the real threats lurking in the digital shadows, making HackO'Lantern a thrilling educational tool for cybersecurity enthusiasts.

- **show\_dialog(self):** Creates and displays spooky dialog boxes with random messages, positions, and opacities.

```
63  def show_dialog(self):
64      self.spooky_beeps()
65      dialog = tk.Toplevel(self.root)
66      dialog.configure(bg='black') # Spooky background color
67      # Get screen width and height
68      screen_width = self.root.winfo_screenwidth()
69      screen_height = self.root.winfo_screenheight()
70
71      # Randomly determine x and y coordinates for the dialog
72      x = random.randint(0, screen_width - 300) # 300 is the width of the dialog
73      y = random.randint(0, screen_height - 100) # 100 is the height of the dialog
74      dialog.geometry(f"300x100+{x}+{y}") # Set the position and size of the dialog
75
76      action = random.choice(['message1', 'message2', 'message3'])
77      spooky_titles = ["Boo!", "Beware!", "Look Behind You!"]
78      dialog.title(random.choice(spooky_titles))
79      if action == 'message1':
80          label = tk.Label(dialog, text=self.spooky_message(), fg="red", bg="black", font=("Chiller", 15))
81      elif action == 'message2':
82          label = tk.Label(dialog, text=self.spooky_message(), fg="blue", bg="black", font=("Chiller", 15))
83      elif action == 'message3':
84          label = tk.Label(dialog, text=self.spooky_message(), fg="white", bg="black", font=("Chiller", 15))
85
86      label.pack(pady=20)
87      button = tk.Button(dialog, text="OK", command=dialog.destroy, bg="grey", fg="white")
88      button.pack(pady=10)
89
90      # Random opacity for ghostly appearance
91      alpha = random.uniform(0.3, 0.7)
92      dialog.attributes('-alpha', alpha)
93
94      # Eerie cursor
95      dialog.config(cursor="pirate")
96
97      self.dialog_count += 1
98
99      if self.dialog_count < 50:
100         self.root.after(5000, self.show_dialog) # Show another dialog after 5 seconds
101     else:
102         self.exit_app(None)
```

The **show\_dialog(self):** function is where the ghostly apparitions of HackO'Lantern come to life, manifesting on the screen to spook the user. This function is responsible for creating and displaying eerie dialog boxes that pop up at random positions on the screen, each bearing a spooky message to unsettle the user.

# HackO'Lantern: Python based scareware

Upon invocation, `show_dialog(self)`: first calls the `spooky_beeps(self)`: function to play a series of creepy beeps, setting an unsettling ambiance. It then conjures a Toplevel dialog from the abyss of the tkinter library, customizing its appearance to be as ghostly as possible. The dialog's background is set to black, and its title is chosen randomly from a list of spooky phrases like "Boo!", "Beware!", and "Look Behind You!".

The function then crafts a spectral message using the `spooky_message(self)`: function, displaying it within the dialog in a chilling red, blue, or white font. The eerie message is accompanied by an "OK" button that allows the user to banish the ghostly dialog with a click.

To enhance the spectral effect, `show_dialog(self)`: sets a random opacity for the dialog, making it appear ethereal and ghost-like. It also changes the cursor to a pirate symbol, adding to the eerie atmosphere.

The function keeps a count of the dialog apparitions using the `dialog_count` attribute. If the count is less than 50, it schedules another invocation of `show_dialog(self)`: after 5 seconds, ensuring a relentless barrage of spooky dialogs to haunt the user's screen. However, the haunting ceases when the count reaches 50, calling `exit_app(None)` to bring the eerie experience to a close.

The `show_dialog(self)`: function encapsulates the essence of HackO'Lantern's spooky interaction with the user, making it a central part of the scareware experience.

- `exit_app(self, event)`: Exits the application when the F9 key is pressed.

```
104     def exit_app(self, event):
105         self.root.quit()
```

- `run(self)`: Initiates the spooky sequence by calling `show_dialog` and `fake_delete_files`, and starts the Tkinter main loop.

```
107     def run(self):
108         self.show_dialog()
109         self.fake_delete_files() # Start the fake file deletion process
110         self.root.mainloop()
```

## SPICING UP THE SPOOKINESS:

Students are encouraged to modify the program to make it more fun or spooky. Here are some suggestions:

- Add more spooky sound effects using the Winsound module.
- Create additional eerie animations or visuals using the tkinter library.
- Enhance the fake file deletion scare by displaying the file paths in the dialog boxes instead of the console.
- Incorporate more keyboard shortcuts for different spooky effects or to exit the application.

Now that you have a deeper understanding of the dark arts behind HackO'Lantern, feel free to explore, experiment, and enhance the spookiness to your heart's content. Remember, the goal is to learn, have fun, and keep the scares **FRIENDLY AND HARMLESS!**

Now, we need to package the program into an executable!

## PACKAGING HACKO'LANTERN: FROM SCRIPT TO EXECUTABLE:

Packaging HackO'Lantern into a standalone executable is the final step in preparing this spooky software for its eerie escapades across different operating systems. This process encapsulates the Python script and all its ghostly dependencies into a single binary file, making it easy to share and run without requiring a Python environment.

Begin by embarking on a spectral search for an icon that encapsulates the eerie essence of HackO'Lantern. There are many online repositories of icons, like [Icon-icon](#), [IconFinder](#), or [Flaticon](#), where you can find a ghastly glyph to represent your application. Once you've chosen an icon, download it in .ico format for Windows or .icns format for OSX.

## HackO'Lantern: Python based scareware

With your icon in hand, it's time to summon the packaging spirits using a tool like PyInstaller or cx\_Freeze. For this example, we'll use PyInstaller for its simplicity and support across Windows, OSX, and Linux.

Install PyInstaller using pip:

```
pip install pyinstaller
```

Now, navigate the command line to the directory containing your HackO'Lantern script. Cast the following incantation to package your script, replacing your-icon.ico with the path to your icon file and HackOLantern.py with the name of your script:

```
pyinstaller --onefile --icon=your-icon.ico HackOLantern.py
```

As the packaging ritual completes, you'll find the resulting executable in the dist directory, now bearing the spooky icon you selected. Your HackO'Lantern is ready to haunt computers with its eerie executable, spreading spooky cybersecurity awareness wherever it roams!

As we draw the curtains on the eerie narrative of HackO'Lantern, it's essential to reflect on the spirit of this spooky endeavor. This project is conjured from the cauldron of creativity and fun, meant to spook and amuse, not to wreak havoc or cause distress. While pranking your friends within a circle of trust can lead to hearty laughs and memorable scares, extending such pranks to unsuspecting souls who are not in on the fun can morph the amusement into distress.

HackO'Lantern is an open crypt for all curious minds. You are encouraged to delve into its code, tweak the spooks, and share your ghostly versions on the sacred grounds of GitHub. By starring in the HackO'Lantern project and making a pull request, you contribute to a community of digital ghostbusters, each with a unique flair for eerie fun.

So, as you venture into the digital haunted house, remember that the essence of HackO'Lantern is to learn, share, and enjoy the spooky spirit of coding, keeping the malicious specters at bay. Let the ghostly camaraderie thrive in the heart of the open-source realm, and may your code be as spooky as the midnight chime on a Halloween night!



WiCyS UArizona Student Chapter Presents:

# 2023 Cybersecurity Speaker Series

## Ashley Burke

Information Security Program Manager, Wave HQ.

**Nov 27<sup>th</sup> - 6:00 p.m.**

Zoom

Open to the public.

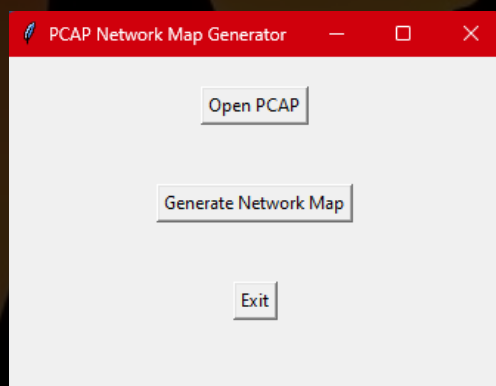
<https://arizona.zoom.us/j/85472545975?pwd=R2R3eXpJRUNrbFk2WTRtS2x0S2VnZz09>



# Enhancing Network Analysis through Automated Network Map Generation

In my course, CYBV 326 - Introductory Methods of Network Analysis, understanding the topology and the interactions within a network is crucial for Network Analysis and lays a strong foundation for understanding the principles and methodologies involved in analyzing networks. As a practical extension to the concepts learned in this course, the PCAP Network Map Generator tool is a valuable asset for network analysts.

The PCAP Network Map Generator is a simple Python-based application developed to automate the process of generating network maps from Packet Capture (PCAP) files. This tool encapsulates the essence of network analysis in a user-friendly graphical interface using libraries such as Scapy for packet analysis, NetworkX for network modeling, and Bokeh for interactive visualization. The program begins by allowing users to upload a PCAP file through a simple "Open PCAP" button. Once a file is selected, the "Generate Network Map" button activates, enabling the user to represent the network interactions captured in the PCAP file visually.

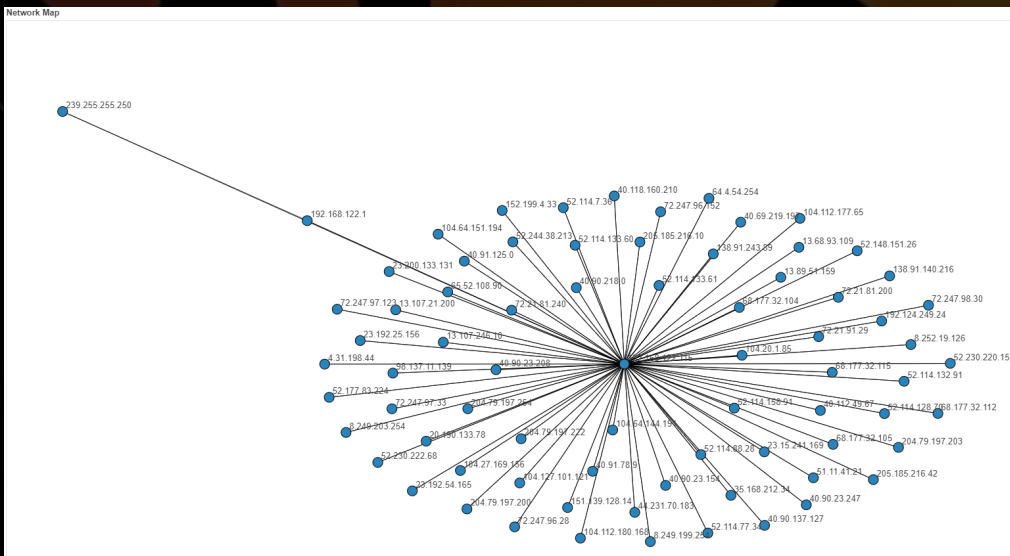


The program reads the PCAP file upon initiation, iterating through each packet to extract source and destination IP addresses. These IP pairs are then used to construct a directed graph, where nodes represent IP addresses and edges represent their interactions. The graph is subsequently visualized in an interactive plot, representing the network's topology and interactions. This visualization is rendered in a web browser, allowing easy navigation and exploration. Additionally, the tool provides an option to save the generated network map as an HTML file for future reference or sharing.

# Enhancing Network Analysis through Automated Network Map Generation

This tool significantly streamlines the understanding of network behavior post-capture for network analysts. By providing a visual insight into the network's structure and interactions, analysts are better equipped to identify patterns, anomalies, or potential areas of optimization. Moreover, the interactive nature of the generated network map facilitates a more intuitive understanding, enabling analysts to delve deeper into the network's dynamics.

The PCAP Network Map Generator tool is a practical bridge between theoretical knowledge acquired in CYBV 326 and real-world network analysis tasks. By automating the tedious process of mapping network interactions from PCAP files, this tool not only saves time but also enhances the analytical capabilities of network analysts, making it a worthy addition to the toolkit of anyone keen on mastering the art and science of network analysis.



First, we will go to this project's GitHub directory by navigating to <https://github.com/mgalde/PCAPMap>. This project has a few requirements to parse network information and generate a graph based on the PCAP.

We will run the following command in pip to install these requirements.

```
pip install scapy networkx bokeh
```



# Enhancing Network Analysis through Automated Network Map Generation

## Importing Necessary Libraries:

The code begins by importing essential libraries:

- tkinter for creating the graphical user interface (GUI).
- scapy for reading and parsing the PCAP file.
- networkx for creating and manipulating the network graph.
- bokeh for visualizing the network graph interactively.
- webbrowser to display the generated network map in a web browser.

```
1 import tkinter as tk
2 from tkinter import filedialog, messagebox
3 from scapy.all import *
4 import networkx as nx
5 from bokeh.io import output_file, show
6 from bokeh.models import Plot, Range1d, Multiline, Circle, HoverTool, BoxZoomTool, ResetTool, LabelSet, ColumnDataSource
7 from bokeh.plotting import from_networkx
8 from bokeh.palettes import Spectral4
9 from bokeh.layouts import layout
10 import webbrowser
```

## Defining the PCAPAnalyzer Class:

The PCAPAnalyzer class inherits from tk.Tk, the base class for creating windows in Tkinter. The constructor (\_\_init\_\_) initializes the window, sets its title, and dimensions, and creates three buttons: Open PCAP, Generate Network Map, and Exit.

```
12 class PCAPAnalyzer(tk.Tk):
13     def __init__(self):
14         super().__init__()
15
16         self.title("PCAP Network Map Generator")
17         self.geometry("800x800")
18
19         self.open_button = tk.Button(self, text="Open PCAP", command=self.open_pcap)
20         self.open_button.pack(pady=20)
21
22         self.generate_button = tk.Button(self, text="Generate Network Map", command=self.generate_map, state=tk.DISABLED)
23         self.generate_button.pack(pady=20)
24
25         self.exit_button = tk.Button(self, text="Exit", command=self.destroy) # Exit button
26         self.exit_button.pack(pady=20) # Adjust padding as needed
```

# Enhancing Network Analysis through Automated Network Map Generation

## Opening the PCAP File:

The `open_pcap` method utilizes `filedialog.askopenfilename` to prompt the user to select a PCAP file. Once a file is selected, it enables the Generate Network Map button.

```
29 def open_pcap(self):
30     self.pcap_path = filedialog.askopenfilename(title="Select PCAP File", filetypes=[("PCAP Files", "*.pcap;*.pcapng")])
31     if self.pcap_path:
32         self.generate_button.config(state=tk.NORMAL)
```

## Generating the Network Map:

The `generate_map` method is the core of this tool. It reads the PCAP file using Scapy's `rdpcap` function, iterates through the packets to extract source and destination IP addresses, and constructs a directed graph using NetworkX. It then relabels the nodes with integer labels for better visualization, creates a plot using Bokeh, and renders the network graph interactively.

```
34 def generate_map(self):
35     try:
36         pcap = rdpcap(self.pcap_path)
37     except Exception as e:
38         messagebox.showerror("Error", f"Failed to read PCAP file: {e}")
39         return
40
41     G = nx.DiGraph()
42
43     for packet in pcap:
44         if IP in packet:
45             src_ip = packet[IP].src
46             dst_ip = packet[IP].dst
47             G.add_edge(src_ip, dst_ip)
48
49     # Relabel nodes with integer labels
50     mapping = {node: i for i, node in enumerate(G.nodes())}
51     G_relabeled = nx.relabel_nodes(G, mapping)
52
53     # Create a new position dictionary with integer keys
54     pos = nx.spring_layout(G, k=0.15, iterations=20)
55     pos_relabeled = {mapping[node]: pos[node] for node in pos}
56
57     plot = Plot(width=1920, height=1080, # Adjust these values to your desired size
58               x_range=Range1d(-1.1, 1.1), y_range=Range1d(-1.1, 1.1))
59     plot.title.text = "Network Map"
60
61     graph_renderer = from_networkx(G_relabeled, pos_relabeled, scale=1, center=(0, 0))
62     graph_renderer.node_renderer.glyph = Circle(size=15, fill_color=Spectral14[0])
63     graph_renderer.edge_renderer.glyph = MultiLine(line_alpha=0.8, line_width=1)
64     plot.renderers.append(graph_renderer)
65
66     plot.add_tools(HoverTool(tooltips=None), BoxZoomTool(), ResetTool())
67
68
69     source = ColumnDataSource(data=dict(
70         x=[pos[0] for pos in pos.values()],
71         y=[pos[1] for pos in pos.values()],
72         label=list(G.nodes())
73     ))
74
75     # Create a LabelSet with the labels from the ColumnDataSource
76     labels = LabelSet(x='x', y='y', text='label', source=source,
77                     text_font_size='10pt', x_offset=5, y_offset=5)
78
79
80     # Add the labels to the plot
81     plot.add_layout(labels)
82
83     plot_layout = layout([plot])
84
85     output_file("network.html")
86     show(plot_layout) # This will save the plot to network.html and open it in the web browser
87
88     self.save_button = tk.Button(self, text="Save Network Map", command=lambda: self.save_map(plot))
89     self.save_button.pack(pady=20)
```

# Enhancing Network Analysis through Automated Network Map Generation

## Saving the Network Map:

The `save_map` method allows the user to save the generated network map as an HTML file using `filedialog.asksaveasfilename` to specify the file path.

```
91 def save_map(self, plot):
92     file_path = filedialog.asksaveasfilename(defaultextension=".html", filetypes=[("HTML Files", "*.html")])
93     if file_path:
94         try:
95             output_file(file_path)
96             show(plot)
97         except Exception as e:
98             messagebox.showerror("Error", f"Failed to save network map: {e}")
```

## Potential Modifications:

- **Error Handling:** Enhance error handling by catching specific exceptions, which could provide more informative error messages to the user.
- **File Validation:** Implement additional validation to ensure the selected file is a valid PCAP file before attempting to process it.
- **Performance Optimization:** Optimize the packet processing loop to handle large PCAP files more efficiently, possibly through parallel processing or by utilizing more efficient data structures.
- **Additional Features:** Introduce features like filtering options to allow users to focus on specific types of network traffic or exporting the network map to different file formats.
- **Styling and Theming:** Enhance the visualization by allowing users to customize the appearance of the network map, such as changing colors, node sizes, or layout algorithms.

Understanding the code structure and functionality allows you to extend and adapt the PCAP Network Map Generator to meet their needs better or explore new avenues in automated network analysis. Through such modifications, you can deepen their understanding of network analysis methodologies, honing their network analysis and software development skills in network analysis and cybersecurity tool development.

# Silken Threats: Unraveling Scattered Spider's Cyber Assaults BY: Professor Michael Galde

In the digital realm, where data is the new gold, nefarious entities often spin intricate webs to ensnare the unwary. Scattered Spider echoes among the rogue gallery of cyber marauders with a chilling resonance across the casino industry. With its penchant for social engineering and sophisticated ransomware attacks, this elusive group has recently cast its sinister web over the glittering facades of MGM Resorts and Caesars Entertainment. The cyber onslaughts orchestrated by Scattered Spider not only unveiled the vulnerabilities nestled within the digital fortresses of these casino giants but also sent a ripple of alarm across an industry that thrives on the trust and engagement of its clientele. As we delve deeper into the dark alleys of these cyber-attacks, we unravel the tactics employed by Scattered Spider, the repercussions faced by the targeted enterprises, and the lessons that the cybersecurity aficionados at the University of Arizona Cyber Operations program can glean from these real-world digital skirmishes. Through the lens of these incidents, we aim to shed light on the ever-evolving landscape of cyber threats and the imperative for robust cybersecurity measures in safeguarding the treasure troves of data held by modern enterprises.

## 1. INCIDENT DISCOVERY AND DISCLOSURE:

- MGM Resorts discovered a cybersecurity issue on September 11, 2023, affecting its main website, online reservations, and in-casino services. The disclosure was made on the same day, showcasing a relatively prompt response to the incident.
- Caesars Entertainment discovered suspicious activity on its network on September 7, 2023, due to a social engineering attack on an outsourced IT support vendor. The disclosure was made on September 14, 2023, following a Bloomberg report, indicating a delay in public disclosure compared to MGM.

## 2. ATTACK VECTOR:

- MGM Resorts faced a ransomware attack by an affiliate of the BlackCat/ALPHV ransomware group, known as Scattered Spider, which encrypted over 100 ESXi hypervisors. This attack vector demonstrates the capability of threat actors to exploit virtualization environments, which are crucial for modern IT operations.
- Caesars Entertainment was breached through an outside IT vendor, which was then used to gain access to the company's network. The same group, Scattered Spider, was behind this attack, showcasing the group's ability to exploit third-party relationships to gain unauthorized access.

## 3. IMPACT:

- MGM Resorts had to shut down certain systems affecting ATMs, credit card machines, slot machines, and online services. This impact demonstrates the wide-ranging consequences of a successful cyber-attack on critical infrastructure within the hospitality and gaming industry.
- Caesars Entertainment had its loyalty program database compromised, containing sensitive information like driver's license and social security numbers. This breach underscores the importance of securing customer data and the potential for identity theft and fraud.

# Silken Threats: Unraveling Scattered Spider's Cyber Assaults on MGM and Caesars BY: Professor Michael Galde

## 4. RESPONSE:

- MGM Resorts shut down affected systems and began an investigation. The company also switched to manual operations to mitigate the impact, showcasing a business continuity plan.
- Caesars Entertainment activated incident response protocols, engaged cybersecurity firms, and notified law enforcement. They also offered credit monitoring and identity theft protection services to affected loyalty program members, indicating a proactive approach to mitigate the potential fallout from the data breach.

## 5. FINANCIAL IMPLICATIONS:

- The financial implications for MGM Resorts are not explicitly mentioned, but the disruption in services likely resulted in financial loss, including potential reputational damage and loss of customer trust.
- Caesars Entertainment paid a ransom in tens of millions of dollars to prevent the leak of stolen data, showcasing the financial pressures companies face when dealing with ransomware attacks.

## 6. THREAT ACTOR:

Both attacks were orchestrated by the group known as Scattered Spider, showcasing their ability to target large enterprises within a similar industry. This highlights the need for industry-specific threat intelligence and collaborative cybersecurity efforts within similar business sectors.

The cyberattacks on MGM Resorts and Caesars Entertainment underscore the critical importance of robust cybersecurity measures, especially in industries handling vast amounts of sensitive customer data. The incidents reveal how threat actors exploit vulnerabilities in the organizations' systems and their extended network of vendors.

These real-world incidents provide a rich learning ground for the cybersecurity students at the University of Arizona Cyber Operations program. By studying the attack vectors, response strategies, and the aftermath of such cyberattacks, students can better understand the challenges enterprises face and the crucial role of cybersecurity professionals in preventing, mitigating, and responding to cyber threats. They will be better equipped to defend organizations against similar cyber threats through rigorous training and education, thereby contributing to a safer digital landscape.

Furthermore, the technical aspects of these attacks, such as the exploitation of ESXi hypervisors and third-party vendor relationships, offer a deep dive into advanced threat vectors, emphasizing the need for a multi-layered security approach and continuous monitoring of both internal and external threat landscapes.

- <https://www.bleepingcomputer.com/news/security/mgm-resorts-shuts-down-it-systems-after-cyberattack/>
- <https://www.bleepingcomputer.com/news/security/caesars-entertainment-confirms-ransom-payment-customer-data-theft/>
- <https://www.sec.gov/Archives/edgar/data/1590895/000119312523235015/d537840d8k.htm>
- <https://finance.yahoo.com/news/caesars-entertainment-paid-millions-ransom-185252473.html>
- <https://www.bleepingcomputer.com/news/security/the-week-in-ransomware-september-15th-2023-russian-roulette/>
- <https://www.bleepingcomputer.com/news/security/mgm-casinos-esxi-servers-allegedly-encrypted-in-ransomware-attack/>

Welcome to the OCTOBER 2023 issue of THE PACKET! As the hues of autumn become more pronounced, so does our journey through the realms of cybersecurity here at the University of Arizona Cyber Operations program. I'm your steadfast companion through this expedition, Professor Michael Galde. We're amidst the bustling Fall semester, with midterms signifying our halfway mark. As we forge ahead, I warmly embrace the seasoned and the newcomers in our academic community.

October brings a chill in the air and the whimsical spirit of Halloween. As you revel in the festivities, I wish you a fun and safe Halloween! Amidst the merriment, let's not let our guard down in the digital sphere, for as the seasons change, so do the tactics of our cyber adversaries.

The cybersecurity horizon was tinged with concern as the University of Michigan recently experienced a cyber attack. Though details remain scant, THE PACKET will thoroughly analyze the incident as more information comes to light. Our quest for understanding the ever-evolving threat landscape continues unabated. I am very excited to break this event down but am disappointed in the lack of communication from the involved parties.

This edition takes a slightly whimsical turn as we delve into more creative hacking projects. This month's hacking projects feature two engaging initiatives: an introduction to Python GUI interface and a template to develop your cybersecurity tools with our network map project and a scareware program called HackO'Lantern. With the job market increasingly valuing technical acumen, these projects are not merely academic exercises but a stepping stone toward enhancing your marketability and prowess in cybersecurity. So, add these to your GitHub and build your profile!

As you navigate through midterms, the spirit of inquiry and the thrill of discovery may be your guiding lights. Immerse yourself, debate vigorously, and continue to quench your thirst for knowledge. Let's continue sculpting an academic year rich in learning, growth, and collaborative triumphs. Stay curious, remain steadfast, and let's voyage through the unknown together!

## CONTACT US

CIIO@EMAIL.ARIZONA.EDU

1140 N. Colombo Ave. | Sierra Vista, AZ 85635

Phone: 520-458-8278 ext 2155

<https://cyber-operations.azcast.arizona.edu/>

