IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF GEORGIA
ATLANTA DIVISION

MICROSOFT CORPORATION,

Plaintiff,

v.

DOES 1-10,

Defendants.

Case No. 1:25-CV-2695-MHC

FILED UNDER SEAL

DECLARATION OF RODELIO FIÑONES IN SUPPORT OF MICROSOFT'S MOTION FOR TEMPORARY RESTRAINING ORDER AND RELATED RELIEF

I, Rodelio Fiñones, declare as follows:

1. I am a Principal Security Software Engineer & Reverse Engineer in Microsoft CELA Cybersecurity & Trust Engineering ("CSTE"); I formerly served in a similar role within Microsoft's Digital Crimes Unit ("DCU"). My scope was expanded to also include AI and its abuse research. I make this declaration based upon my personal knowledge, and upon information and belief from my review of documents and evidence collected during Microsoft's investigation.

2. I have been employed by Microsoft since June 2009. In my role at Microsoft, I assess technological security threats to Microsoft and the impact of such threats on Microsoft's business and customers. I work with a team of

investigators that focuses in part on researching different categories of malware, including botnets. My team and I research emerging malware threats through analysis of submitted samples, reverse engineering, forensic examination, data stream analysis, and development of tools to track botnet development. I am the team lead in developing malware prevention and eradication tools. Prior to joining Microsoft, I worked from 2004-2009 for Fortinet Technologies (Canada), Inc. as a Principal Software Developer/ Researcher (2007-2009) and Senior Antivirus Analyst (2004-2007). My job included research and analysis of complex malware and the development of tools to detect and eradicate malware. From 1999-2004, I worked for Trend Micro, Inc. as a Senior Anti-Virus Researcher and Anti-Virus Engine Developer. During my professional career, I have received advanced, specialized training and extensive practical experience in investigating malware and botnets and in devising technical countermeasures to detect and disable them. A true and current copy of my curriculum vitae can be found in **Exhibit 1**.

3. I regularly create, analyze, use, and reverse engineer software as part of my duties at Microsoft. Much of my work includes source code analysis. Source code refers to human-readable text for instructing a computer how to perform a function. When a software author sets out to write computer instructions in source code form, they have many options for how to express the instructions in

human readable form. Factors that a software author may consider in determining how to express instructions in source code form include the ease with which another human can understand the code, the manner in which the code is organized, the way the code being authored relates to other portions of source code that it may interact with, security, and the ease of compiling the code into binary form (1's and 0's) for interpretation and execution by computers.

4. Since approximately September 2024, I have been part of the team investigating malware known as the Lumma, LummaC2, or LummaStealer malware ("Lumma"). My role to date has included observation, testing, and reverse engineering of the Lumma malware. My role has also included investigation into the Microsoft software and systems affected by the Lumma malware in order to understand how the malware leverages Microsoft's Windows software. My role has also included working with other investigators at Microsoft, including my colleagues Derek Richardson and Igor Aronov.

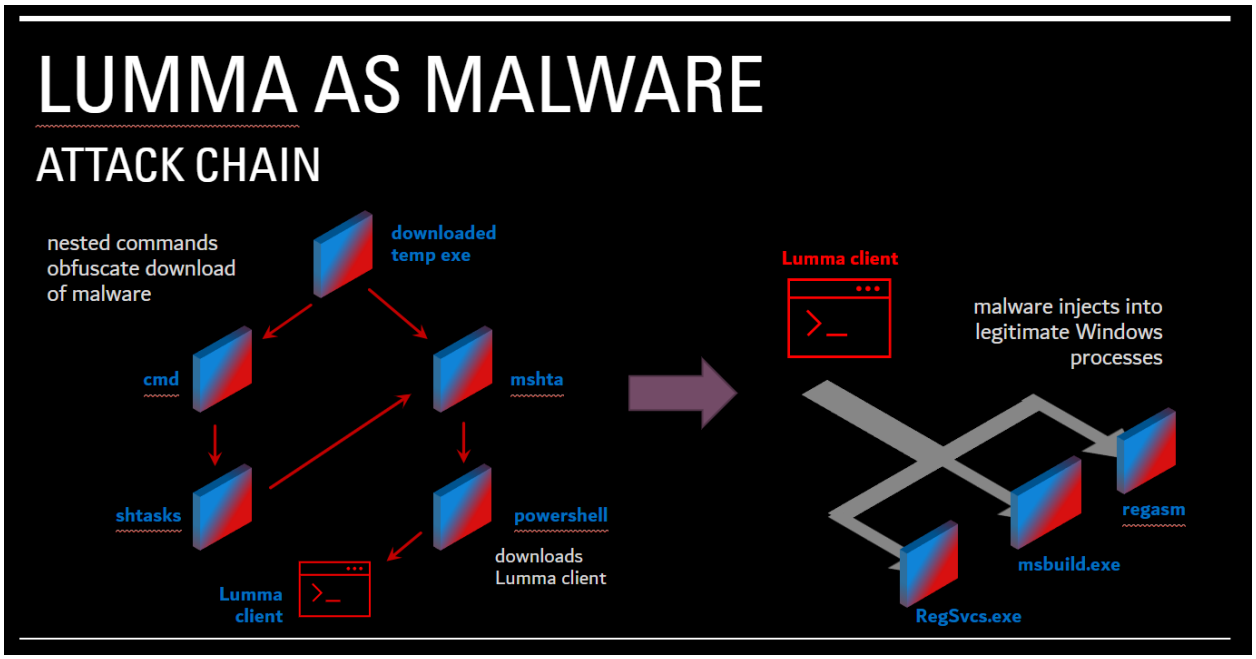
5. Microsoft analyzed Lumma malware using a combination of reverse engineering techniques, including dynamic and static reverse engineering methodologies. Dynamic reverse engineering refers to running malware samples in a secured and controlled environment that permits tracking and logging of malware behaviors. Microsoft performed dynamic reverse engineering using debugger tools

like IDA and Windbg, which allowed investigators to step through malware codes as they are executed in the controlled environment. Microsoft also performed static reverse engineering that involved unpacking malware code with tools like IDA that permit investigators to read partially decompiled code and other malware artifacts.

6. Lumma malware is written in the C++/ASM programing language. The code is heavily obfuscated, which indicates an attempt by the authors of the malware to make it difficult for Microsoft and other security researchers to understand its functions. Obfuscation methods employed by the authors of Lumma include low-level virtual machine compiler infrastructure (LLVM) and control flow flattening (CFF). LLVM refers to virtual machine code compilers that use modular design and transformation capabilities to obfuscate code as it is compiled from source code to binary code. CFF refers to restructuring program control flow to make it harder to follow what a program is doing.

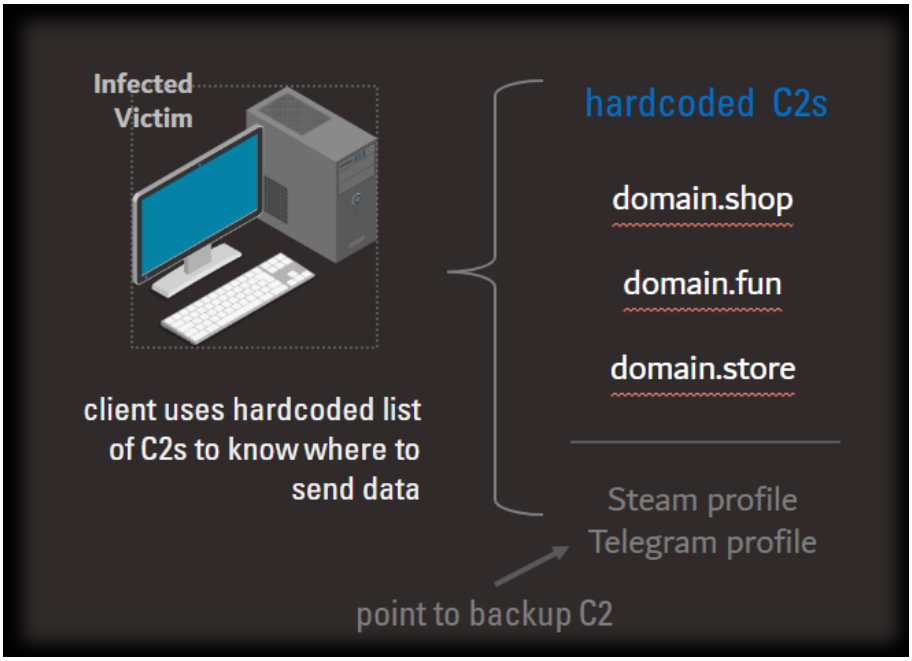
7. The Lumma attack chain begins with social engineering tactics designed to trick victims into installing Lumma malware on their computer, as discussed in the declaration of Derek Richardson. Once a user has been tricked into installing Lumma, a series of nested commands are used to obfuscate the download of malware files and bypass Microsoft security measures. These files infect the victim computer and make it a "client" of the Lumma network. The

Lumma malware injects malicious commands into legitimate Windows processes to avoid detection. **Figure 1** below depicts the Lumma attach chain at a high level.

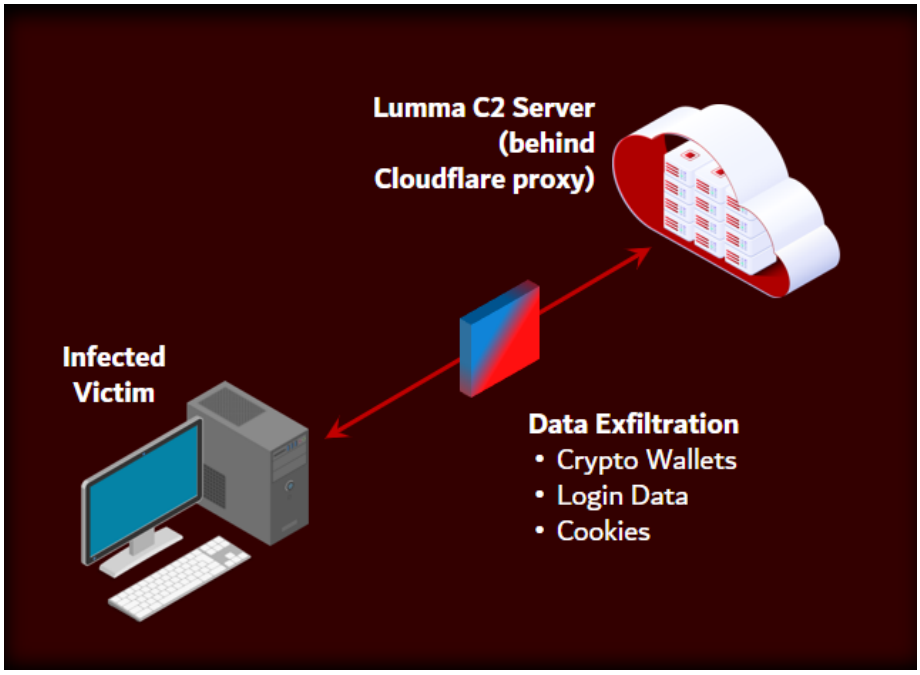


8. Once a victim computer is infected, Lumma can exfiltrate data from the infected computer for transmission to command and control servers operated by bad actors like the Defendants in this case. The ".data" section of the Lumma malware contains multiple hard-coded domains that infected computers can communicate with by default. There are also Telegram channels hard-coded into the ".data" section of Lumma that infected computers can reach out to to obtain C2 domains. Microsoft observed the Telegram channels to be the first C2 domain to be called when present. In addition to the hard-coded Telegram channels, there are

also hard coded Steam profiles that may be called by infected computers to retrieve C2 domain information. Steam profiles are hard-coded in the ".text" section of the Lumma code. **Figure 2** below depicts the set of C2 sources observed in Lumma.



9. Defendants make use of proxy servers to obfuscate the location of the actual command and control servers, as depicted in **Figure 3** below.

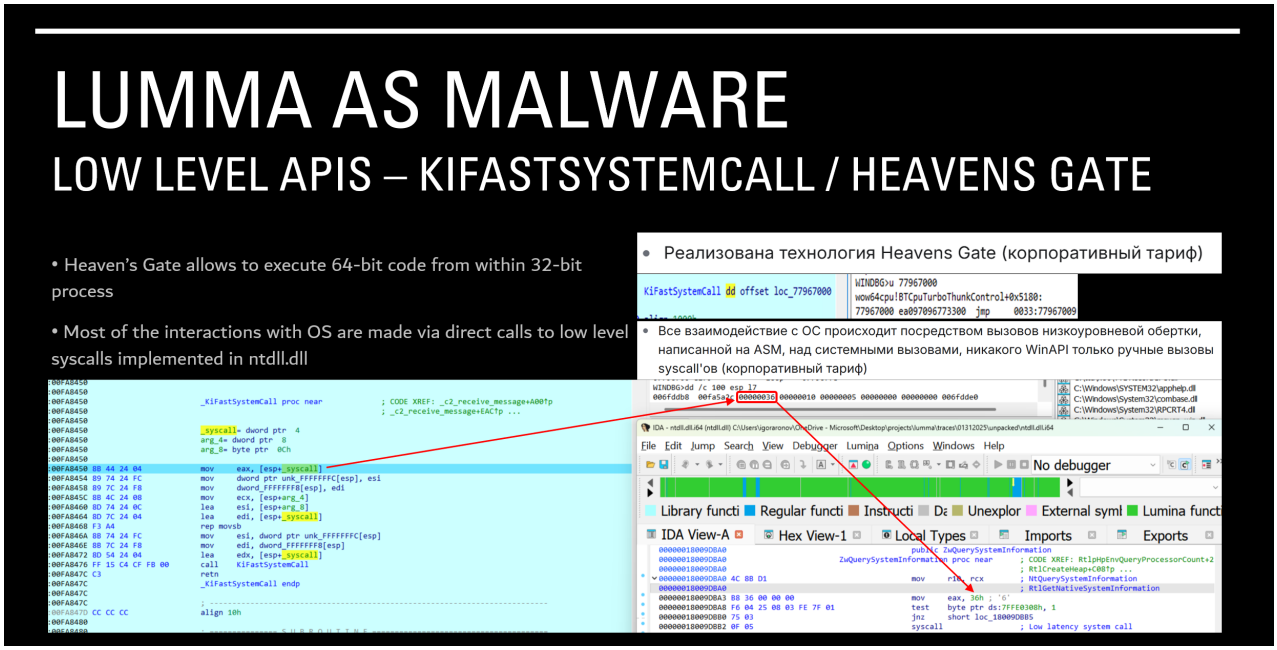


10. Analysis of Lumma shows that the malware targets several types of victim information including user's files (documents from under %userprofiles%), credentials from browsers (login data like username, passwords and credit card numbers, search histories, web data, user and network cookies), crypto wallets and extensions, two factor authentication web browser extensions, and data associated with VPN, FTP, and email applications. For example, if Edge browser is open and Lumma attempts to steal browser cookies, it will terminate processes related to Edge and will restart the process with specific command line as if it attempts to debug it.

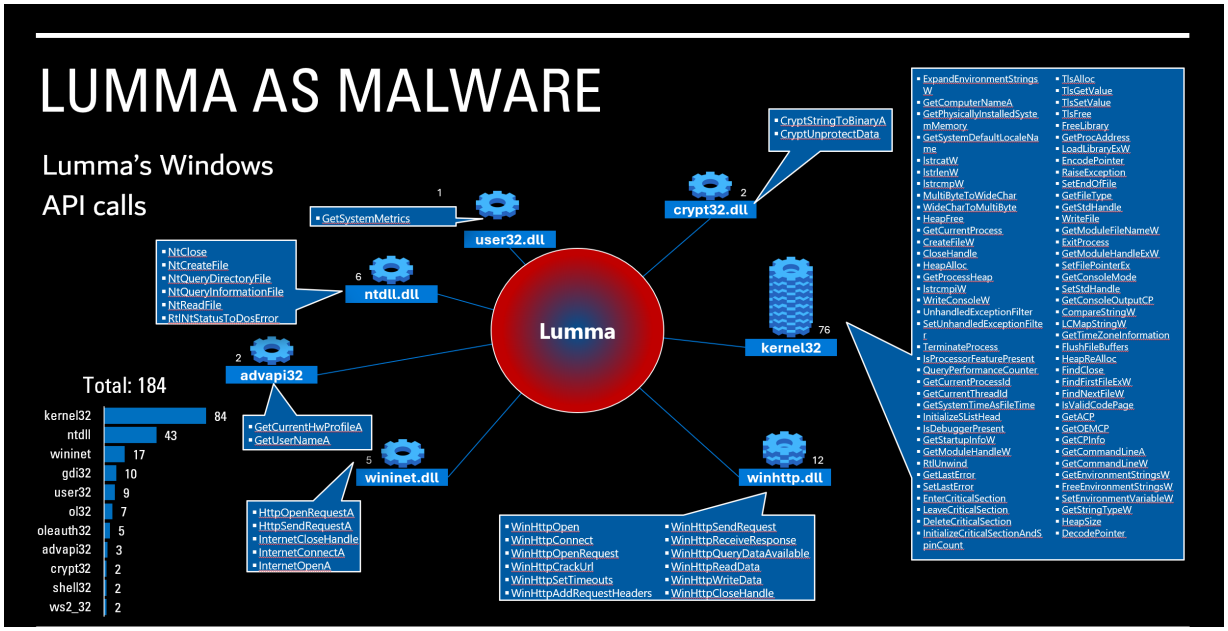
11. Lumma also makes use of something commonly known as the "Heavens Gate" technique. This technique is designed to exploit a feature in

operating systems that allows the transition from a 32-bit mode to a 64-bit mode during execution of certain software. The Heavens Gate technique enables malware to evade detection by security software designed to monitor [1]Lumma's ntdll.dll. Analysis of Lumma also shows that it is designed to bypass Microsoft antivirus products. The malware attempts to install a driver that is used by a malicious loader program to deliver/download Lumma and other malware modules. Lumma malware also attempts to terminate services related to Microsoft Security Products**.** The snapshots at **Figure 4** below shows the execution flow.

[1] An API, or application programing interface, is computer code that allows software applications to communicate with each another. At a high level, an API consists of two types of code: declaring code, which is the code used to call a function, and implementing code, which is the code that actually performs a function once it has been called. I understand that both declaring code and implementing code can be subject to copyright protection.



12. Lumma malware makes at least 184 Windows API calls during the course of its operation. Windows APIs are APIs created by Microsoft that can be used to facilitate communications between the Windows operating system and third-party software applications. **Figure 5** below provides a graphical display of the APIs and number of calls observed during Microsoft's investigation:



13. The APIs depicted in Figure 5 belong to Microsoft. I understand that many of these APIs are subject to copyright protection, including in Microsoft's U.S. Copyright Registration No. TX 8-888-365 for the Windows SDK. The declaring code used by Lumma's authors to achieve the calls depicted in Figure 4 equates to many lines of code.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct to the best of my knowledge, information, and belief. Executed this 13th day of May at Atlanta, Georgia.

/s/ Rodelio Fiñones

Rodelio Fiñones

EXHIBIT 1

Rodelio Fiñones

Objective:
Security Engineer/Reverse Engineer with more than 20 years' experience seeking to acquire a rewarding career in the field of cybersecurity where my expertise, experience, and knowledge in cyber threats and software engineering will be utilized to make big impact to the company and Internet ecosystems.

PROFESSIONAL EXPERIENCE:

Principal Security Engineer/Reverse Engineer, Digital Crimes Unit (Nov 2017 – Present)
Microsoft

- Comprehensive reverse engineering of different malwares types.
- Analysis, decryption, and dissecting network captures.
- Design and develop tools and automation systems for analyzing and tracking botnets.
- Utilize big data to hunt known and unknown threats, create IOCs and improve detection/remediation capabilities.
- Collaborate with DCU investigators, lawyers, and internal/external partners to eradicate/disrupt botnets through civil, criminal referrals, or pure blocking strategy.
- Research and develop malware emulators, crawlers, and sinkhole systems for tracking their infrastructures.
- Utilize Azure cloud technologies to accelerate development of robust tools and pipelines to combat cyber threats.
- Prepare and submitted botnet legal declarations for civil and criminal referrals. Some of my declarations below.
    - Dorkbot: *https://botnetlegalnotice.com/dorkbot/*
    - Gamarue: *https://www.noticeofpleadings.net/gamarue/*
    - Trickbot: *https://noticeofpleadings.com/trickbot/*

Senior Antivirus Researcher / Strategist (June 2009 – Nov 2017)
Microsoft

- Handle malware samples from different sources to provide analysis, tracking, detections, advice, and remediation.
- Tracking top acute threats impacting customer and doing end to end research and providing up to date protection and mitigation (Ex: Zeus/Zbot, Locky, Cerber, etc.)

- Lead the team on focus research of different categories of malwares such Botnets, Click fraud, MSIL, and Spambots to provide different kind customer protection such as sourcing, protections (File, memory, and cloud-based, behavior-based, etc.). Also includes identifying the malware infection chain and monetization.
- Design and develop tools and automation projects for unpacking and tracking botnets.
- Utilize big data (cosmos) to analyze malware treat landscape and to better protect the customer.
- Lead team and hands on research on prevalent malwares for CME (MS Collective Malware Eradication) to disrupt/eradicate malwares.
- End to end research and development of antimalware engine and product features to improve customer protections.
- Drive improvements to windows platform and its components (OS, script engines – JS, PS, etc.)
- Static and dynamic analysis of different types of malwares and vulnerabilities.
- Creating rules and programs to improve automatic detections of malwares.
- Respond to malware outbreaks.
- Write tweets, blogs, research papers, and present to top security conferences.
- Provide mentoring to other researchers.

Principal Software Developer / Researcher (Dec 2007 – June 2009) Fortinet Technologies (Canada), Inc.

- Improved signature-based detection algorithm to support more complex malwares. Detection methods such as x-raying and behavioral and characteristics detections.
- Research and develop AV engine features to support packer through script based. Generic unpacker coupled with script-based unpacker will be a powerful and effective solution for most malwares.
- Handle complex malwares analysis and detection through script-based or hardcoded.
- Improvements and optimizations for Fortinet's AV detection algorithms.
- Provide technical knowledge to new AV analyst.

Senior Antivirus Analyst / Engine developer (May 2004 – Dec 2007) Fortinet Technologies (Canada), Inc.

<u>Analysis, Research and Development Projects</u>

- Research, design, and develop the clean engine for Fortinet's desktop Antivirus product. It supports cleaning Win32 PE, Office, DOS, and script formats.

- Improve the scanning technology through research and development of new scanning algorithm that suits for complex viruses.
- Fix bugs and AV scan and clean engine limitations
- Participate in the research, development, and improvements of Win32 emulator engine.
- Create detection module for hard to detect viruses such metamorphic, polymorphic, and EPO viruses.
- Improved the scanning technology for scripts malwares.
- Research, design, and develop an automated system to replicate, analyze, and heuristically detect known and unknown malwares through sandboxing technology.
- Handle complex malwares.
    - Analysis
    - Creating detection signatures
    - AV Engine support (if necessary)

Other Tasks:

- Provide malware related technical expertise to analyst and product development team.
- Create documentation for developed systems.
- Conduct trainings regarding virus analysis, detection, and disinfection algorithm.
- Provide quick and quality solution to customer problems.
- Configure replication system for any kinds of malware

Senior Anti-Virus Researcher / AV Engine Developer (Nov 1999 – April 2004) Trend Micro, Inc. (Anti- Virus and Internet Security)

AV Trainee

 3-month extensive virus / malware training. Include analysis and creation of detection and clean signatures.

AV Technical Support Engineer

 Provides complete solution to the customer.

- Provides scan and clean solution. Includes signature to remove system infections such as registry, system files, process, services, and files.
- Create detailed / comprehensive virus description and manual removal instructions.
- Provides other assistance needed by the clients.

AV Research Engineer

- Focus on the detailed / comprehensive analysis of Windows viruses.
- Process escalation cases from Virus support engineers.

- Conduct technology upgrade trainings to Virus support team (New virus technology; new Scan / Clean feature).
- Respond to Virus Alerts
- Develop removal tools for specific malware.
- Design and develop TSC (Trojan System Cleaner). Engine module to detect and restore system infection through registry, process, system files, and services.
- Process Scan / Clean Engine related cases.
- Analyze exploits and system vulnerabilities (Windows & Linux).
- Research and develop a system for automating the replication of malware that covers controlled and simulated Internet environment Research and develop Scan/Clean Engine modules:
    - Metamorphic virus support
    - Win32 virus clean modules
    - New file formats support
    - Trojan System Cleaner
    - Compression / Packer engine support (UPX, Petite, and PEPack)

TECHNICAL SKILLS:

- Advance knowledge and experience in reverse engineering any kinds of malware using debugger (Soft-ice, IDA Pro, OllyDbg, and Immunity debugger)
- In-depth knowledge and experience in exploits and vulnerabilities.
- Strong knowledge and experience in creating comprehensive malware description.
- Strong experience in developing detection and cleaning engine for different kinds of malware such as virus, worms, Trojans, and spywares.
- In-depth knowledge in windows operating system internals.
- In-depth knowledge and experience in virus, Trojans, worms, and spywares behaviors.
- Experience in TCP/IP networks, Unix/Linux networks (AIX, Redhat, Slackware, Ubuntu), Windows network (Windows 9X/NT/2K), Novell Netware. Background knowledge in Windows CE, Palm, and EPOC operating system.
- Advance experience in C and DOS/Win32 Assembly, VB Script, JavaScript.
- Experience in Linux shell scripts, PowerBuilder, and SQL programming.
- Intermediate experience in analyzing and testing various Anti-virus products.
- Intermediate experience and knowledge in network protocols like TCP/IP, IPX/SPX, SMTP, FTP, HTTP, DNS, NTTP, and MAPI32.
- Intermediate knowledge in Windows and Unix/Linux system and network security.
- Knowledge in ASP, MS Access, FoxPro, and HTML.
- Knowledge in IP chains/tables, Ethereal packet sniffer, Snort IDS, Tripwire integrity checker.
- Knowledge in Lotus Notes and Domino server

- Advance experience in Python and Django web framework.
- Advance experience in software development utilizing Azure cloud technologies.
- Advance experience in Network forensics.
- Advance experienced in software development using C#, Python, and C/C++.
- GIAC Certified Incident Handler
- GIAC Advisory Board

PREVIOUS EMPLOYMENT:

System analyst / Programmer (July 1999, October 1999)
Gestalt Consulting Inc. (PowerBuilder and MS SQL)

- Create, maintain, and improve Inventory system and Accounting system.
- Provide support to problem of clients.
- Review and document the current application system.

EDUCATION:

Bachelor of Science in Computer Engineering (1995 -1999)
FEU - East Asia College of Information Technology, May 1999