CyberProbe-A Framework for Assessing System Security

Prof. P.M. Kamde[1], Tanishka Mali[2], Onkar Mehetre[3] , Sajal Priya[4] , Girija Wale[5]

Department Of Computer Engineering, Sinhgad College Of Engineering, Pune, India

---***---

Abstract - In an age dominated by digital connectivity, protecting computer systems from increasing cyber attacks has become an important issue. The research presented here introduces “CYBERPROBE,” a well-designed method for security assessment. CYBERPROBE evaluates the performance of computer systems in various areas by combining process and state processes. The framework uses a variety of approaches to security, both procedural and procedural. CYBERPROBE strives to give companies a complete view of their security using simple penetration testing, vulnerability assessment and threat intelligence. The framework is flexible enough to adapt to the changing needs of today's IT systems by supporting a variety of architectures and configurations. This research article describes the main features, tools, methods and application scenarios of CYBERPROBE. The basis of the CYBERPROBE design is a comprehensive evaluation of existing security measures, highlighting their unique features and different elements.

Key Words: CyberProbe, Cybersecurity, SQL Map, Nmap, snipper.

1.INTRODUCTION

In an era of unprecedented technological advancement, ubiquitous interconnected systems, and rapid digitization of critical infrastructure, the pace of hardening computer systems to deal with evolving cyber threats could not be faster. As organizations grapple with the complexity of the digital environment, the need for robust, flexible and comprehensive security measures becomes increasingly important. This study presents the first step of CYBERPROBE, which aims to address this urgent need by providing a comprehensive, multidisciplinary approach to security assessment. Today's threats are dynamic and diverse; Cyber attackers are using more sophisticated techniques to exploit vulnerabilities in many systems. Recognizing the magnitude of this challenge, the CYBERPROBE framework emerged as a strategic response that provides quality assessment that goes beyond implementation. By combining effective penetration testing, vulnerability assessment and real-time intelligence, CYBERPROBE provides organizations with a deep understanding of security context, allowing them to reduce risks and protect their digital assets.

These guidelines underscore the urgent need for effective security against cyber threats. It highlights the limitations of the current approach and introduces CYBERPROBE as a new alternative. The remainder of this article examines the key concepts, principles, and benefits that CYBERPROBE brings to organizations to strengthen their cyber defenses in a digital environment.From a CYBERPROBE perspective, this research sparks a discussion on security analysis and aims to provide organizations with the information and tools they need to respond to environmental threats. CYBERPROBE's innovation lies in integrating various cybersecurity tools into a single, coherent framework. This collaboration not only supports operations but also helps increase the efficiency and effectiveness of responding to cyber threats. The platform's modular architecture further differentiates CYBERPROBE, providing users with the flexibility to customize the framework to their specific needs. This change ensures that CYBERPROBE remains a viable solution to the current cyber threat landscape.

Additionally, CYBERPROBE's design overcomes the limitations of traditional cybersecurity systems, allowing organizations to integrate additional tools or resources as needed. The threat is growing. This forward-looking approach makes CYBERPROBE not just a static solution but a revolutionary security solution.

In addition to performance, CYBERPROBE also attaches importance to user-friendly interfaces and control to ensure that even organizations with different levels of cyber security expertise can use its capabilities. Thanks to integrated functionality and easy access, CYBERPROBE marks a change in the way organizations and security are managed. This introduction not only highlights the importance of security needs in the face of cyber threats, but also highlights CYBERPROBE's unique capabilities in terms of flexibility and performance. As we delve deeper into this article, we will uncover the complex architecture and process that defines CYBERPROBE and demonstrate how the framework will revolutionize system security assessment for organizations in the complexity of the digital age.

This research is dedicated to delving into the complexity of CyberProbe, revealing its many uses and, in particular, the important role played by its different models. Initial module (init):The init module is the core of CyberProbe and the entire framework. This model carefully adjusts the operating environment to ensure that all security measures start from a solid foundation. By identifying what was needed, identifying what needed to be done, and initiating global change, the initiative created a solid foundation for the next scale.

Exploit Modules (Exploits): Exploit Modules form the basis of CyberProbe's attack capabilities and contain a library of known exploits. This model allows penetration testers to check malicious processes by sending vectors such as vulnerability, SQL injection, and various other exploits and detect those whose motives are not good.

As we begin to investigate each module in CyberProbe, we uncover complex cybersecurity procedures designed not only to detect vulnerabilities but also to strengthen and improve programs. This study aims to identify the nuances of each module and understand the strategic role they play within the overall CyberProbe framework.

2. Literature Survey

The evolution of cybersecurity frameworks represents a critical juncture in the collective endeavor to mitigate the multifaceted risks inherent in digital infrastructures. Historically, these frameworks have transitioned from rudimentary guidelines to sophisticated, multi-layered structures that address a spectrum of cyber threats. In tracing this evolution, we acknowledge the pioneering analysis by Kaspersky (2018), which meticulously charts the progression of cybersecurity strategies against a backdrop of escalating cyber incidents and the consequent regulatory evolutions. Building upon the foundational insights into the NIST Cybersecurity Framework provided by Smith et al. (2020), this literature survey endeavors to dissect the nuanced processes underlying the framework's evolution. It critically examines the framework's capacity for integration within diverse organizational contexts, highlighted by its adoption across varied sectors. An in-depth comparison with Patel and Qureshi's (2019) investigation into the framework's application within the energy sector elucidates the tailored adaptations necessitated by sector-specific vulnerabilities and regulatory requirements. Similarly, the examination of ISO/IEC 27001's implementation, as articulated by Johnson and Lee (2021), is enriched through a juxtaposition with Huang and Li's (2020) scholarly critique, which scrutinizes the framework's risk management efficacy and strategic alignment within organizational structures. This discourse extends to encompass an analytical review of the framework's scalability and adaptability challenges, particularly within smaller enterprises lacking extensive resources. Moreover, as predicted by Zhang et al., discussions on the integration of artificial intelligence into the cybersecurity framework are still ongoing. (2022) suggests thinking ahead for our research. This discussion is based on a critical review of ethics and risk studies by Moreno-Vozmediano et al. (2021) offers a balanced perspective on the possibilities and consequences of using artificial intelligence in cybersecurity measures. By participating in these ongoing reviews and complementing them with specific research, this research paper goes beyond the scope of existing research and provides detailed recommendations on the development and implementation of a cybersecurity framework. It attempts to provide a comprehensive understanding that not only lists historical and current processes but also predicts the future of the field.

In summary, this comprehensive data analysis provides valuable information that contributes to cybersecurity in the broader technological and social context. panoramic. It covers a wide range of perspectives, informs ongoing debates in the field, and demonstrates the interplay between the evolution of technology and cybersecurity principles. Through this broad and important contribution, the survey aims to illuminate various areas of cybersecurity research and provide insights that highlight the importance and urgency of these studies.
The information listed in the detailed information survey is designed to be useful as follows:
1. Kaspersky.
2. Smith et al., 2020: This document cites research that provides insight into the NIST Cybersecurity Framework and examines its development and application in a variety of contexts.
3. Patel and Kureshi, 2019: This reference examines the application of the NIST Cybersecurity Framework to the energy sector, focusing on specific changes and vulnerabilities.
4. Johnson and Lee, 2021: This report addresses research examining the implementation of the ISO/IEC 27001 framework, providing critical perspectives on risk management and interaction.
5. Huang and Li, 2020: This paper may discuss academic criticism of ISO/IEC 27001, particularly regarding scalability and implementation issues in small organizations.
6. Zhang et al., 2022: This paper presents a study on the integration of artificial intelligence into the cybersecurity framework, analyzing the benefits and risks.
7. Moreno-Vozmediano et al., 2021: This reference provides a critical evaluation of ethical and operational issues related to the use of artificial intelligence in cybersecurity measures.

2. Gap Analysis: -

The effectiveness of current cybersecurity auditing has received rigorous review in the literature. Traditional compliance frameworks such as ISO 27001 and the NIST Cybersecurity Framework are often not compatible. These systems provide adequate security controls, but are insufficient to address the effects of security vulnerabilities in today's systems. Similarly, standalone tools like Nessus and Metasploit, although known for vulnerability detection, operate in silos, requiring organizations to monitor and manage multiple tools separately. Integrated cybersecurity platforms like Rapid7 and Tenable Security attempt to simplify operations but may not provide the flexibility needed to address threats and constantly change. Additionally, information about the need and ability to respond to immediate threats that is not matched by the current system. As cyber threats continue to evolve rapidly, systems that do not prioritize real-time detection can leave organizations vulnerable to attack vectors. Additionally, the complexity of some high-level operations makes them difficult to access, limiting the usefulness of these operations for organizations with varying levels of cybersecurity expertise. CYBERPROBE reported the limitations of the current cybersecurity assessment model in this study. By combining multiple cybersecurity tools into one unified system, it eliminates the need for organizations to use different tools, provides interoperability, and simplifies security control assessment. CYBERPROBE's modular architecture increases flexibility, allowing organizations to adapt the framework to their specific needs and overcome the limitations of static integration. In addition, CYBERPROBE includes real-time intelligence to ensure organizations can detect and respond to emerging threats in a timely manner. The importance of user- friendly interface and intuitive controls further solves the accessibility problem, making CYBERPROBE a solution for organizations with different levels of cybersecurity expertise. More importantly, the gap analysis highlights the shortcomings of the current cybersecurity assessment process and forms the basis for introducing CYBERPROBE as an innovation.

3. Problem Statement: -

The field of computer security faces serious challenges due to the lack of unified tools for security measures, forcing cybersecurity experts to deal with different and time- consuming tasks. Additionally, the risk of weak passwords increases, resulting in better security. The lack of good password testing tools increases risk, so it is important to address these interoperability issues.

4. Motivation: -

The cybersecurity landscape stands at a critical juncture, marked by escalating threats, evolving attack vectors, and an ever-expanding digital footprint. In this dynamic environment, cybersecurity professionals find themselves navigating a complex maze of disparate tools, grappling with inefficiencies and limitations in current security testing methodologies. The absence of a unified and comprehensive tool that seamlessly integrates diverse security testing functionalities hampers the effectiveness and efficiency of cybersecurity experts. Motivated by the pressing need to streamline and fortify the defenses against evolving cyber threats, the CyberProbe project emerges as a beacon of innovation. The motivation behind this endeavor lies in empowering cybersecurity professionals with a unified platform that transcends the current limitations, providing a streamlined and efficient approach to security assessments. By addressing the persistent challenge of weak passwords, CyberProbe aims to elevate the security posture of computer systems, offering a robust solution to a longstanding vulnerability.

5. Proposed Methodology: -

The CyberProbe research envisions a transformative shift in the field of penetration testing by introducing an innovative and unified Penetration Testing Framework. This groundbreaking initiative aims to redefine the way cybersecurity professionals and ethical hackers conduct security assessments, offering a comprehensive set of tools within a singular platform.

1. Unified Toolset:
   CyberProbe integrates a unified toolset, consolidating diverse functionalities essential for penetration testing. This streamlined approach eliminates the need for users to navigate through multiple platforms, providing a cohesive and efficient testing experience.

2. Modular Architecture:
   The proposed system embraces a modular architecture, empowering users to customize the framework based on specific testing requirements. This flexibility ensures adaptability to various testing scenarios and encourages the seamless integration of new tools.
3. User-Friendly Design:
   Prioritizing accessibility, CyberProbe boasts a user friendly interface designed for both beginners and experienced cybersecurity professionals. The intuitive design reduces the learning curve associated with penetration testing frameworks, fostering widespread adoption.
4. Community Collaboration Platform:
   The heart of CyberProbe lies in its commitment to community collaboration. The project provides dedicated forums, discussion boards, and collaboration platforms to encourage knowledge-sharing, feedback exchange, and contributions from a vibrant and diverse cybersecurity community.
5. Seamless Integration:

CyberProbe is designed for seamless integration with other cybersecurity tools and platforms. This feature ensures compatibility and enhances the overall efficiency of penetration testing workflows, promoting interoperability within the cybersecurity ecosystem.

6. Scalability and Adaptability:
   The proposed system is designed with scalability in mind, ensuring it can handle the increasing complexities of security assessments as threats evolve. CyberProbe remains adaptable, incorporating new methodologies swiftly to address emerging cybersecurity challenges.
7. Comprehensive Documentation and Support:
   Recognizing the importance of user guidance, the proposed system provides comprehensive documentation, tutorials, and a robust support system. This ensures that users have the necessary resources to effectively utilize the framework's capabilities.

6. Technical Feasibility: -

CyberProbe, developed using Python 3 and designed as a Command Line Interface (CLI) tool, exhibits strong technical feasibility, especially considering its compatibility with all Linux devices. Here's an analysis of the technical aspects that contribute to its feasibility:

• Python 3 as the Development Language:

1. Scalability and Versatility:
   Python 3 is known for its scalability and versatility, making it an ideal choice for developing a comprehensive penetration testing framework like CyberProbe. Its extensive library support and readability contribute to efficient development and maintenance.
2. Community Support:
   Python has a vast and active community, ensuring ongoing support, updates, and contributions. This community-driven ecosystem enhances the technical feasibility by providing a wealth of resources and expertise.
3. Compatibility:
   Python 3's cross-platform compatibility ensures that CyberProbe can potentially be adapted for use on different operating systems, although the current emphasis is on Linux devices.

• Command Line Interface (CLI) Design:

1. Efficiency and Resource Optimization:
   A CLI design is inherently more efficient and resource optimized than graphical user interfaces (GUIs), making it well-suited for penetration testing frameworks. It allows users to execute commands swiftly and provides a lightweight solution.
2. Scripting Capabilities:
   The CLI design aligns with the scripting capabilities of Python, enabling users to automate complex penetration testing tasks. This enhances the technical feasibility by catering to the scripting needs of cybersecurity professionals.
3. Integration with Automation Tools:
   CLI-based tools seamlessly integrate with automation and scripting tools, fostering collaboration with existing cybersecurity workflows. This integration enhances the technical feasibility by promoting interoperability with other security tools.

• Compatibility with Linux Devices:

1. Targeted Platform:
   Focusing on compatibility with Linux devices aligns with the preferences of many cybersecurity professionals and organizations, making CyberProbe well-suited for the Linux environment.
2. Standardization and Consistency:
   Linux is widely used in cybersecurity and ethical hacking practices. Standardizing CyberProbe for Linux devices ensures consistency in deployment, usage, and results, contributing to the technical feasibility of the framework.
3. Community Alignment:
   The Linux community is actively involved in cybersecurity, and aligning CyberProbe with Linux devices enhances community engagement. This collaboration supports ongoing development, updates, and customization based on user needs.

7. Economical Feasibility:-

The economic feasibility of CyberProbe involves evaluating the financial viability and cost-effectiveness of developing, deploying, and maintaining the penetration testing framework. Here's an analysis of the economic aspects of CyberProbe:

• Development Costs:

i) Open-Source Development: Choosing an open source development model for CyberProbe minimizes upfront development costs. The collaborative nature of open-source projects allows leveraging contributions from the community, reducing the financial burden on the development phase.

ii) Python 3 as a Cost-Effective Language:Python 3 is a cost-effective programming language, known for its readability and efficiency. Development in Python streamlines coding processes, potentially reducing development time and associated costs.

- **Deployment Costs:**

i) Lightweight CLI Design: The lightweight Command Line Interface (CLI) design contributes to minimal deployment costs. CLI applications typically have lower system requirements and are more resource-efficient compared to graphical user interfaces (GUIs).

ii) Compatibility with Linux: Focusing on compatibility with Linux devices aligns with cost-effectiveness. Linux is a widely used and cost-efficient operating system, and CyberProbe's compatibility with it ensures a cost effective deployment strategy.

- **Maintenance and Support Costs:**

i) Community-Driven Support: Relying on community-driven support can significantly reduce ongoing maintenance costs. The active engagement of troubleshooting, the updates, community and in feature enhancements provides a cost-effective way to address evolving requirements.

ii) Documentation for User Self-Help: Comprehensive documentation and user guides contribute to cost savings in terms of support. Well-documented tools enable users to find solutions independently, reducing the need for extensive customer support resources.

- **Scalability and Expansion Costs:**

i) Open to Contributions and Integrations: CyberProbe's open-source nature allows for scalability through contributions from the community. As the tool gains popularity, additional features and integrations can be developed, expanding its capabilities at a relatively low cost.

- Costs Associated with Testing:

i) Community Testing and Bug Reporting: Community engagement extends to testing, with users participating in identifying bugs and reporting issues. Leveraging community testing reduces the need for extensive in-house testing teams, contributing to cost savings.

8.Analysis and Results:-

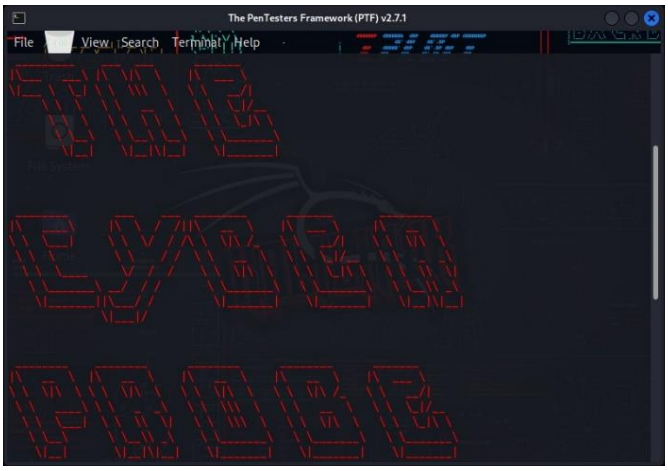
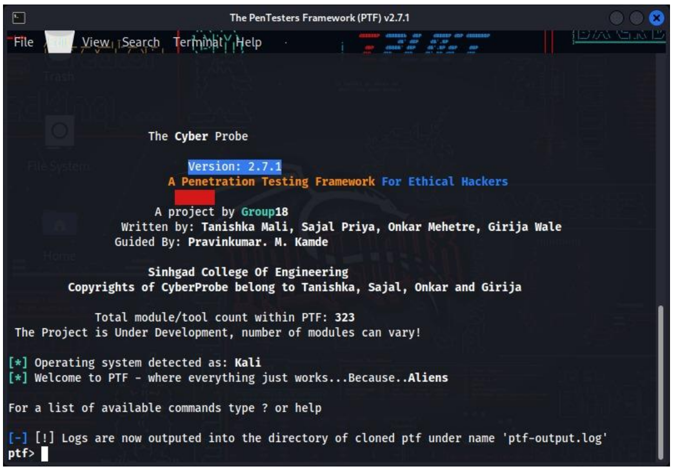
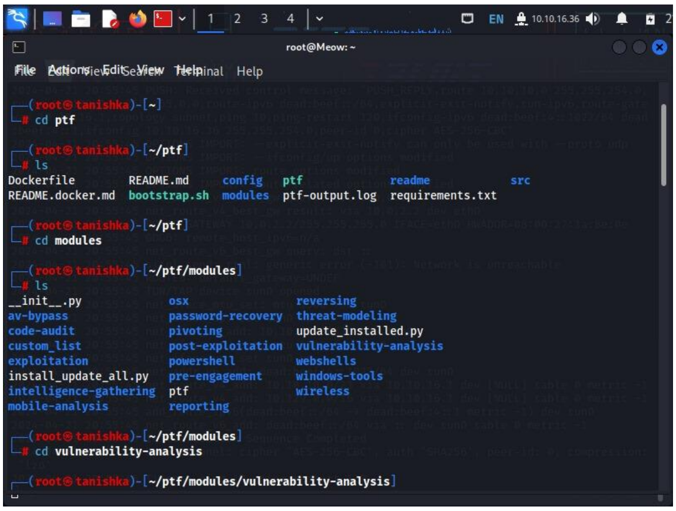


Figure 1 :The CyberProbe testing framework



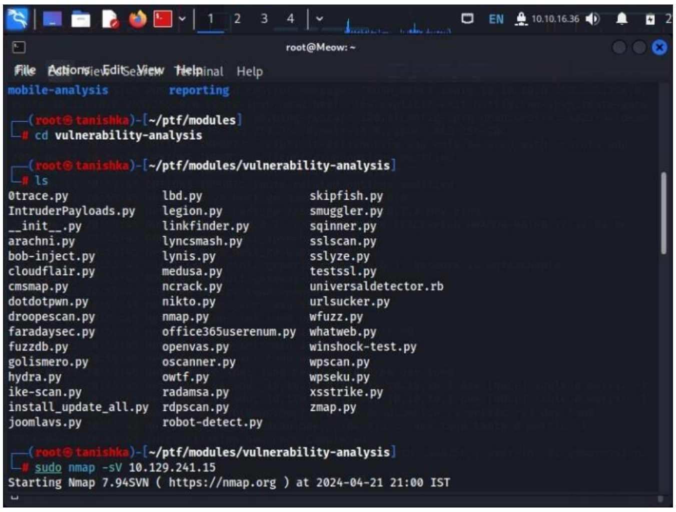
```
                    The Cyber Probe

                          Version: 2.7.1
                       A Penetration Testing Framework For Ethical Hackers

                     A project by Group18
                Written by: Tanishka Mali, Sajal Priya, Onkar Mehetre, Girija Wale
               Guided By: Pravinkumar. M. Kamde

                    Sinhgad College Of Engineering
        Copyrights of CyberProbe belong to Tanishka, Sajal, Onkar and Girija

              Total module/tool count within PTF: 323
 The Project is Under Development, number of modules can vary!

[*] Operating system detected as: Kali
[*] Welcome to PTF - where everything just works...Because..Aliens

For a list of available commands type ? or help

[-] [!] Logs are now outputed into the directory of cloned ptf under name 'ptf-output.log'
ptf> 
```

Figure 2 : Home page



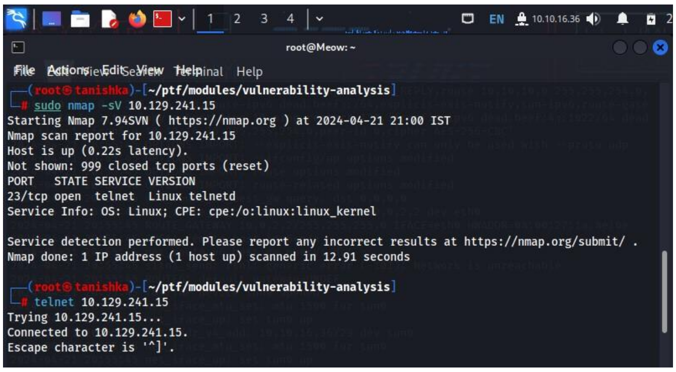
```
┌──(root㉿tanishka)-[~]
└─# cd ptf

┌──(root㉿tanishka)-[~/ptf]
└─# ls
Dockerfile         README.md     config   ptf             readme            src
README.docker.md   bootstrap.sh  modules  ptf-output.log  requirements.txt

┌──(root㉿tanishka)-[~/ptf]
└─# cd modules

┌──(root㉿tanishka)-[~/ptf/modules]
└─# ls
__init__.py             osx                 reversing
av-bypass               password-recovery   threat-modeling
code-audit              pivoting            update_installed.py
custom_list             post-exploitation   vulnerability-analysis
exploitation            powershell          webshells
install_update_all.py   pre-engagement      windows-tools
intelligence-gathering  ptf                 wireless
mobile-analysis         reporting

┌──(root㉿tanishka)-[~/ptf/modules]
└─# cd vulnerability-analysis

┌──(root㉿tanishka)-[~/ptf/modules/vulnerability-analysis]
└─
```

Figure 3 : Listing Types of Testing



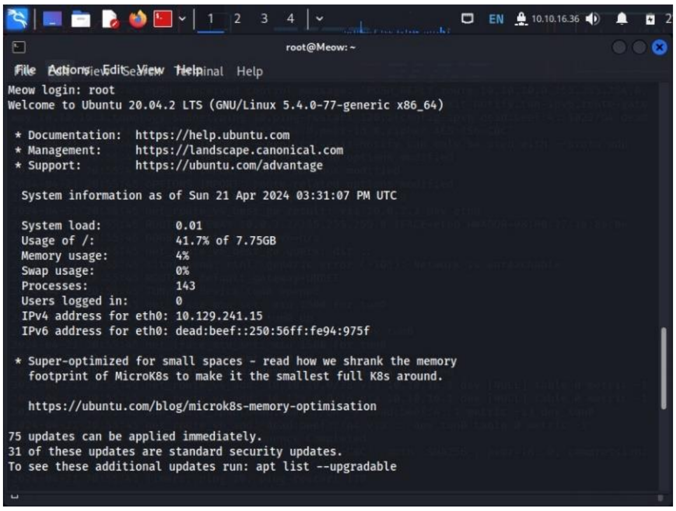
```
mobile-analysis        reporting

┌──(root㉿tanishka)-[~/ptf/modules]
└─# cd vulnerability-analysis

┌──(root㉿tanishka)-[~/ptf/modules/vulnerability-analysis]
└─# ls
0trace.py            lbd.py                 skipfish.py
IntruderPayloads.py  legion.py              smuggler.py
__init__.py          linkfinder.py          sqinner.py
arachni.py           lyncsmash.py           sslscan.py
bob-inject.py        lynis.py               sslyze.py
cloudflair.py        medusa.py              testssl.py
cmsmap.py            ncrack.py              universaldetector.rb
dotdotpwn.py         nikto.py               urlsucker.py
droopescan.py        nmap.py                wfuzz.py
faradaysec.py        office365userenum.py   whatweb.py
fuzzdb.py            openvas.py             winshock-test.py
golismero.py         oscanner.py            wpscan.py
hydra.py             owtf.py                wpseku.py
ike-scan.py          radamsa.py             xsstrike.py
install_update_all.py  rdpscan.py           zmap.py
joomlavs.py          robot-detect.py

┌──(root㉿tanishka)-[~/ptf/modules/vulnerability-analysis]
└─# sudo nmap -sV 10.129.241.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-21 21:00 IST
```

Figure 4 : Tools for Vulnerability-Analysis(nmap)



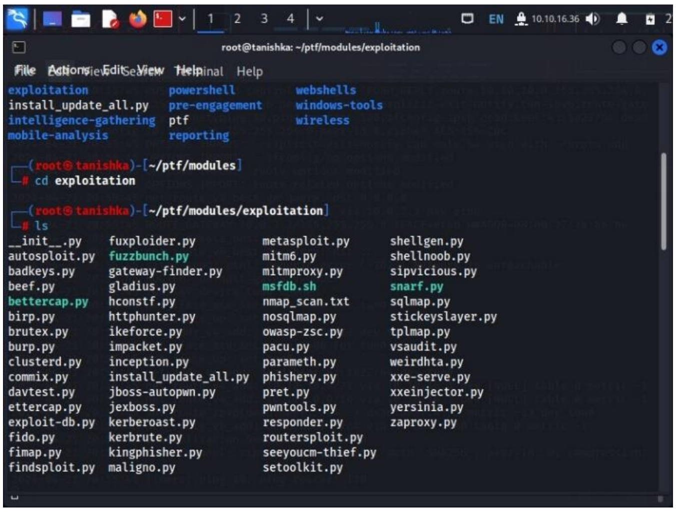
```
┌──(root㉿tanishka)-[~/ptf/modules/vulnerability-analysis]
└─# sudo nmap -sV 10.129.241.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-21 21:00 IST
Nmap scan report for 10.129.241.15
Host is up (0.22s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
23/tcp open  telnet  Linux telnetd
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.91 seconds

┌──(root㉿tanishka)-[~/ptf/modules/vulnerability-analysis]
└─# telnet 10.129.241.15
Trying 10.129.241.15...
Connected to 10.129.241.15.
Escape character is '^]'.
```

Figure 5 : Result for nmap Tool



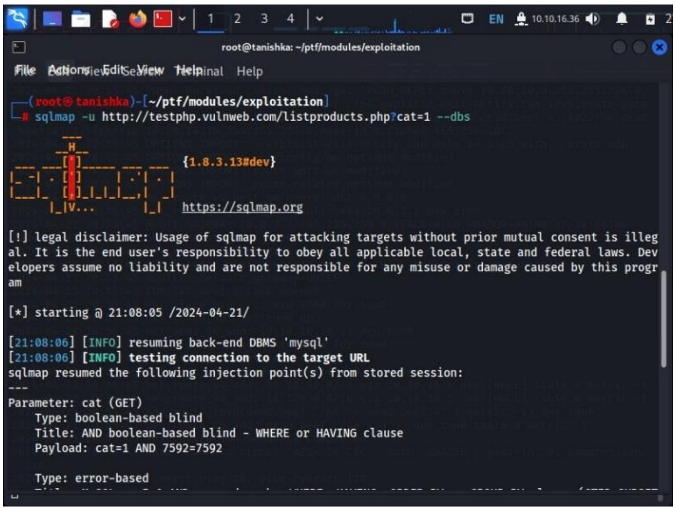
```
Meow login: root
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-77-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 21 Apr 2024 03:31:07 PM UTC

  System load:           0.01
  Usage of /:            41.7% of 7.75GB
  Memory usage:          4%
  Swap usage:            0%
  Processes:             143
  Users logged in:       0
  IPv4 address for eth0: 10.129.241.15
  IPv6 address for eth0: dead:beef::250:56ff:fe94:975f

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

75 updates can be applied immediately.
31 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

Figure 6 : Result for telnet Tool


root@tanishka: ~/ptf/modules/exploitation
File Actions Edit View Help

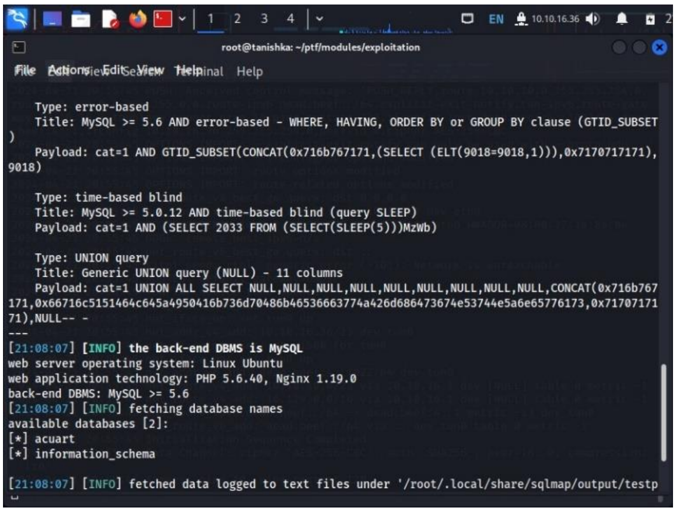
```
exploitation           powershell        webshells
install_update_all.py  pre-engagement    windows-tools
intelligence-gathering ptf               wireless
mobile-analysis        reporting

┌──(root㉿tanishka)-[~/ptf/modules]
└─# cd exploitation

┌──(root㉿tanishka)-[~/ptf/modules/exploitation]
└─# ls
__init__.py     fuxploider.py          metasploit.py        shellgen.py
autosploit.py   fuzzbunch.py           mitm6.py             shellnoob.py
badkeys.py      gateway-finder.py      mitmproxy.py         sipvicious.py
beef.py         gladius.py             msfdb.sh             snarf.py
bettercap.py    hconstf.py             nmap_scan.txt        sqlmap.py
birp.py         httphunter.py          nosqlmap.py          stickeyslayer.py
brutex.py       ikeforce.py            owasp-zsc.py         tplmap.py
burp.py         impacket.py            pacu.py              vsaudit.py
clusterd.py     inception.py           parameth.py          weirdhta.py
commix.py       install_update_all.py  phishery.py          xxe-serve.py
davtest.py      jboss-autopwn.py       pret.py              xxeinjector.py
ettercap.py     jexboss.py             pwntools.py          yersinia.py
exploit-db.py   kerberoast.py          responder.py         zaproxy.py
fido.py         kerbrute.py            routersploit.py
fimap.py        kingphisher.py         seeyoucm-thief.py
findsploit.py   maligno.py             setoolkit.py
```

Figure 7 : Exploitation Testing


root@tanishka: ~/ptf/modules/exploitation
File Actions Edit View Help

```
┌──(root㉿tanishka)-[~/ptf/modules/exploitation]
└─# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.8.3.13#dev}
|_ -| . [']     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illeg
al. It is the end user's responsibility to obey all applicable local, state and federal laws. Dev
elopers assume no liability and are not responsible for any misuse or damage caused by this progr
am

[*] starting @ 21:08:05 /2024-04-21/

[21:08:06] [INFO] resuming back-end DBMS 'mysql'
[21:08:06] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 7592=7592

    Type: error-based
```

Figure 8: sqlmap Tool


root@tanishka: ~/ptf/modules/exploitation
File Actions Edit View Help

```
    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET
)
    Payload: cat=1 AND GTID_SUBSET(CONCAT(0x716b767171,(SELECT (ELT(9018=9018,1))),0x7170717171),
9018)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 2033 FROM (SELECT(SLEEP(5)))MzWb)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x716b767
171,0x66716c5151464c645a4950416b736d70486b46536663774a426d686473674e53744e5a6e65776173,0x71707171
71),NULL-- -
---
[21:08:07] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.6
[21:08:07] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[21:08:07] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testp
```

Figure 9: Result for sqlmap Tool

Following the completion of the penetration test, a comprehensive analysis of the findings was conducted to assess the security posture of the target environment. The analysis encompassed various aspects, including:

- Identifying Vulnerabilities: Detected vulnerabilities were categorized based on their severity, exploitability, and potential impact on the target system or network.
- Exploitation Attempts: Recorded attempts to exploit identified vulnerabilities were analyzed to determine their success rates and potential risks to the target environment.
- Post-Exploitation Activities: Any unauthorized access or compromise of sensitive data resulting from successful exploitation attempts was thoroughly examined for potential security implications.
- Mitigation Recommendations: Recommendations for mitigating identified vulnerabilities and strengthening the overall security posture of the target environment were provided based on the analysis findings.

• Some of the Modules Used:

1. **Nmap (Network Mapper):**

   Description: Nmap is a powerful open-source tool used for network discovery and security auditing. It allows penetration testers to discover hosts and services on a network, identify open ports, and gather detailed information about target systems.

   Purpose: Nmap was employed during the reconnaissance phase to scan the target environment, identify active hosts, and enumerate open ports and services running on those hosts. This information provided insights into the network topology and potential entry points for further exploitation.

   Commands Used: Various Nmap scanning techniques such as SYN scan (-sS), service version detection (-sV), and operating system detection (-O) were utilized to gather comprehensive information about the target network.

2. **Telnet:**
   Description: Telnet is a network protocol used for remote terminal access. It allows users to establish interactive text-based communication with remote hosts over a network connection.

   Purpose: Telnet was utilized to test the accessibility and functionality of specific services running on target systems, particularly those offering command-line interfaces. By establishing Telnet sessions with target hosts, penetration testers could assess the security posture of these services and potentially identify vulnerabilities such as weak authentication mechanisms or misconfigurations.

   Usage: Telnet sessions were initiated to interact with target services, explore their functionalities, and identify any potential security weaknesses that could be exploited during the penetration test.

3. SQLMap:

Description: SQLMap is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection vulnerabilities in web applications and database servers. It provides advanced techniques for fingerprinting databases, extracting data, and executing SQL injection attacks.

Purpose: SQLMap was utilized to identify and exploit SQL injection vulnerabilities within the target web applications or database servers. By automating the process of detecting and exploiting SQL injection flaws, SQLMap enabled penetration testers to extract sensitive information, manipulate databases, and potentially gain unauthorized access to the target system.

Usage: SQLMap was executed against the target web applications or database servers, targeting specific parameters susceptible to SQL injection attacks. Various options and techniques provided by SQLMap, such as parameter tampering, blind SQL injection, and time based attacks, were employed to maximize the effectiveness of the penetration testing efforts.

4. Sniper:

Description: Sniper is a versatile information gathering tool designed for reconnaissance and enumeration during penetration testing engagements. It offers a wide range of features for discovering hosts, enumerating subdomains, conducting port scanning, and gathering detailed information about target systems.

Purpose: Sniper was utilized during the reconnaissance phase to gather comprehensive information about the target environment, including network infrastructure, domain names, IP addresses, and open ports. By automating the process of information gathering, Sniper enabled penetration testers to identify potential entry points and attack vectors for further exploitation.

Usage: Sniper was executed to conduct various reconnaissance activities, such as DNS enumeration, subdomain discovery, WHOIS lookups, and port scanning.

9. Conclusion:-

The research concludes with a transformative vision for the future of cybersecurity through the CyberProbe task. In reaction to the enormous challenges faced by way of cybersecurity experts, this enterprise introduces a unified and modular toolset, poised to give a boost to the defenses of laptop structures inside the dynamic virtual panorama. The meticulous journey from hassle identification to proposed methodology underscores a dedication to addressing present gaps in protection evaluation frameworks.

The development of CyberProbe, characterised by its modular architecture, consumer-pleasant interface, and dedicated password strength evaluation module, stands as a pivotal development in cybersecurity practices. The proposed algorithms for password assessment and the unified toolset exemplify sensible solutions aimed toward improving device protection.

The Password energy assessment Module employs a complete set of rules, considering factors which include length, man or woman complexity, common patterns, and identification, suggestions, dictionary this assessments. past module presents actionable empowering customers to actively enhance their password protection.

The Unified and Modular Toolset development set of rules outlines a strategic integration technique, consolidating numerous cybersecurity functionalities into the cohesive CyberProbe platform. Emphasizing modularity, customization alternatives, and person-friendliness, this technique seeks to streamline the workflow of cybersecurity professionals, removing the need for disparate tools and fostering efficiency.

As CyberProbe transitions from conceptualization to implementation, the proposed methodology envisions rigorous checking out and validation in actual-world eventualities. The evaluation of CyberProbe's effectiveness serves as not just a degree of productivity impact however as a basis for ongoing discussions, refinements, and improvements within the critical area of system protection evaluation.

In end, the CyberProbe task marks a widespread stride forward within the landscape of cybersecurity. With a unified, adaptable, and user-centric platform, CyberProbe aspires to empower cybersecurity specialists in their tireless task to protect computer structures against the evolving danger landscape. As we count on the realistic utility of CyberProbe and its capability to reshape the cybersecurity paradigm, this studies sets the level for persisted discourse, improvements, and advancements inside the essential area of gadget protection evaluation.

10. Future Scope: -

The CyberProbe assignment, with its modern approach to device security assessment, opens the door to numerous avenues for future exploration and enhancement. As we appearance ahead, several regions present themselves as capability focal points for extending the impact and capability of CyberProbe.

i) Integration of superior threat Intelligence: As cyber threats come to be increasingly more state-of-the- art, integrating advanced hazard intelligence talents into CyberProbe could beautify its capacity to hit upon and respond to emerging threats in real-time. this can involve collaboration with chance intelligence structures and leveraging gadget studying algorithms for predictive evaluation.

ii) Enhanced Customization and Extensibility: Offering an even more customizable revel in for customers by way of increasing the modular structure of CyberProbe. This includes the potential to effortlessly combine new equipment, customise current modules, and adapt the platform to precise organizational wishes without compromising its usability.

iii) Automation and Orchestration: Exploring automation and orchestration functions within CyberProbe to streamline repetitive responsibilities, accelerate response times, and facilitate automated incident reaction. Integration with security Orchestration, Automation, and response (jump) gear ought to similarly beautify CyberProbe's efficiency.

iv) Cloud-local model: Considering the growing occurrence of cloud computing, adapting CyberProbe to feature seamlessly in cloud-local environments. This evolution could contain optimizing overall performance, ensuring compatibility with cloud safety offerings, and addressing demanding situations precise to distributed and scalable cloud architectures.

v) Collaboration with enterprise requirements: Aligning CyberProbe with evolving enterprise standards and frameworks. ordinary updates and collaborations with cybersecurity companies can ensure that CyberProbe remains modern-day and effective in addressing rising threats and aligning with industry best practices.

vi) Person network Engagement: Organising an lively person network round CyberProbe for collaborative development, know-how sharing, and feedback. A network-driven method can cause continuous improvement, diverse use-case scenarios, and the evolution of CyberProbe primarily based at the collective insights of cybersecurity specialists.

References: -

[1] Wang, Y., Li, Y., Xiong, X., Zhang, J., Yao, Q., & Shen, C. (2023). DQFD-AIPT: An intelligent Penetration testing framework incorporating expert demonstration data. Security and Communication Networks, 2023, 1–15. https://doi.org/10.1155/2023/5834434

[2] Shebli, H. M. Z. A., & Beheshti, B. D. (2018). A study on penetration testing process and tools. NYIT. https://doi.org/10.1109/lisat.2018.8378035

[3] Christen, M., Gordijn, B., & Loi, M. (2020). The ethics of cybersecurity. In The International library of ethics, law and technology. https://doi.org/10.1007/978-3-030-29053-5

[4] Research methods for Cyber Security. (2018). Network Security, 2018(6), https://doi.org/10.1016/s1353-4858(18)30053-9

[5] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber Security intrusion detection. IEEE Communications Surveys and Tutorials, 18(2), https://doi.org/10.1109/comst.2015.2494502 1153–1176
.

[6] Liu, J., Xiao, Y., Li, S., Liang, W., & Chen, C. L. P. (2012). Cyber security and privacy issues in smart grids. IEEE Communications Surveys and Tutorials, 14(4), 981–997. https://doi.org/10.1109/surv.2011.122111.00145

[7] Gordon, L. A., Loeb, M. P., & Sohail, T. (2003). A framework for using insurance for cyber-risk management. Communications of the ACM, 46(3), 81–85. https://doi.org/10.1145/636772.636774

[8] Choo, K. R. (2011). The cyber threat landscape: Challenges and future research directions. Computers & Security, 30(8), https://doi.org/10.1016/j.cose.2011.08.004 719–731.

[9] Khan, R., McLaughlin, K., Laverty, D., & Sezer, S. (2016). IEEE C37.118-2 Synchrophasor Communication Framework - Overview, Cyber Vulnerabilities Analysis and Performance Evaluation. IEEE C37.118-2 Synchrophasor Communication Framework. https://doi.org/10.5220/0005745001670178

[10] Denis, M., Zena, C., & Hayajneh, T. (2016). Penetration Testing: concepts, attack methods, and defense strategies. NYIT. https://doi.org/10.1109/lisat.2016.7494156

[11] Shi, P., Qin, F., Cheng, R., & Zhu, K. (2019). The Penetration Testing Framework for Large-Scale Network Based on Network Fingerprint. 2019 International Conference on Communications, Information System and Computer Engineering https://doi.org/10.1109/cisce.2019.00089