Practical Cybersecurity toolkit using Python

Akhtar Lodgher, Izzat Alsmadi

Texas A&M University – San Antonio

Goal

- Develop a practical cybersecurity toolkit by undergraduate students using an educational framework
- Students learn and practice foundational cybersecurity concepts
- Students learn to solve problems for applications in network analysis, forensics, system information etc
- Bridge the gap between theoretical knowledge and hands-on practice
- Use cybersecurity related packages of Python
- Goal is for students to understand solution processes of cybersecurity problems, not to develop commercial software to compete with established cybersecurity tools

Design:

Use a simple menu-based framework

- The toolkit consists of multiple modules that implement cybersecurity concepts
- Modules and options are connected through a simple menu-based system
- Easy to understand and implement
- Robust across many platforms (Windows, Linux, Android, etc)
- Expandable
- Use python and various libraries related to cybersecurity

```
Main Menu
---------
1. Network Tools
2. Forensics Tools
3. Get Web Data
4. Get CISA Data
5. Utility Tools
6. Quit Application
Enter integer value of Menu choice (between 1 and 6):
```

Key Features

- Port Scanning
- File Integrity Check
- Password Hashing and Cracking
- Meta data extraction
- Vulnerability assessment
- Large file handling
- Accessing system information

Port Scanning

- Port Scanning using several techniques
- Socket, Nmap, Concurrency

```
Network Tools Menu
------------------
1. Port Scanning using Socket
2. Concurrent Port Scanning using Socket
3. Port Scanning using NMap
4. Return to Main Menu
Enter integer value of Menu choice (between 1 and 4):1
```

```
Port Scanning Menu
------------------
1. Add PortNumber, Add IP Address, Set Connection Timeout, Set LogTime Zone format
2. View: PortNumber, IP Address, Connection Timeout, Port Scan Log, Port Scan Log Time Zone
3. Port Scan IP(s)
4. Save Port Scan Log
5. Clear PortNumber, IP Address, Connection Timeout, Port Scan Log
6. Return to Network Tools Menu
Enter integer value of Menu choice (between 1 and 6):
```

Port Scanning

- Nmap scan variations

```
NMap Scan IPs Menu
------------------
1. Start NMap TCP SYN-ACK Scan on Ports of IPs (unprivileged)
2. Start NMap TCP SYN (Stealth) Scan on Ports of IPs
3. Start NMap UDP Scan on Ports of IPs
4. Start NMap TCP Scan with OS and version on Ports of IPs
5. Start NMap UDP Scan with OS and version on Ports of IPs
6. Start NMap TCP Null Scan on Ports of IPs
7. Start NMap TCP FIN Scan on Ports of IPs
8. Start NMap TCP XMas Scan on Ports of IPs
9. Start NMap TCP ACK Scan on Ports of IPs
10. Start NMap TCP Window Scan on Ports of IPs
11. Start Ping Scan on External IPs (unprivileged). IPs in CIDR notation.
12. Start Ping Scan on Internal IPs. IPs in CIDR notation.
13. Start Ping ACK Scan on External IPs (unprivileged). IPs in CIDR notation.
14. Start Ping ACK Scan on Internal IPs. IPs in CIDR notation.
15. Start Ping UDP Scan on Internal IPs. IPs in CIDR notation.
16. Return to NMap Scanning Menu
Enter integer value of Menu choice (between 1 and 16):
```

Port Scanning

- Concurrent Scanning and data collection

```
Concurrent Port Scan IPs Menu
-----------------------------
1. Start Concurrent Scannning and disply to screen while Scanning
2. Start Concurrent Scannning and Save to CSV while Scanning
3. Return to Concurrent Port Scanning Menu
Enter integer value of Menu choice (between 1 and 3):
```

File Integrity Check

- Use hashing techniques to detect duplicate / tampered files

```
9. Compare two Files for Equality using Hashing
10. Get All Equal files in a directory using Hashing
11. Return to Forensics Tools Menu
Enter integer value of Menu choice (between 1 and 11):
```

Password Hashing and Cracking

- Use PBKDF2 (Password-Based Key Derivation Function), Argon2 and other techniques for secure password storage and cracking experiments. PBKDF2 and Argon2 are both password-based key derivation functions (KDFs) used for password hashing

```
Password Cracker Menu
---------------------
1. Password Cracker Using SHA1 Hash
2. Password Cracker Using SHA256 Hash
3. Password Cracker Using Plain Text
4. Display various hashes of a Password
5. Return to Utility Tools Menu
Enter integer value of Menu choice (between 1 and 5):
```

Category: Linux (with salt):

Method	Password	Salt	HshLen	Rounds	Chk	Hash
DES:	GoJaguars123#	12	13		True	12wMVT7Q1NCDM
bcrypt	GoJaguars123#	$2b$12$6LtfqgDrjSwaIz5UzDZyne60			True	$2b$12$6LtfqgDrjSwaIz5UzDZynePmnv6lJNuQAItI/Hu4OmXmdTq1x5qfO
SunMD5Crypt:	GoJaguars123#	12345678	49	1000	True	$md5,rounds=1000$12345678$$LsL8sVT9M3HicibFKjHWj/
MD5+Salt Hlib:	GoJaguars123#	12345678	32		True	bb48f628fefda318b4771cc3ef025666
MD5+Salt PLib:	GoJaguars123#	12345678	34		True	$1$12345678$YjzZnB45Edus9BLGZsZxl0
SHA1+Salt Hlib:	GoJaguars123#	12345678	40		True	12b6eff960adedc56fab5c0206c1c136c7274564
SHA1+Salt PLib:	GoJaguars123#	12345678	48	1000	True	$sha1$1000$12345678$TEaWWrlkFH.zGuI9Bw66nbNj8GMH
SHA256+Salt Hlib:	GoJaguars123#	12345678	64		True	a6ee0c61474a0096c1941be09bdf9a373c3db2c70d1df38738bbc0f45cbd5af1
SHA256+Salt PLib:	GoJaguars123#	12345678	67	1000	True	5rounds=1000$12345678$nRijgsQDZmQAHqyIXr0.1n5ZkLtsGz2D4SQ42nxQ4a3
SHA512+Salt Hlib:	GoJaguars123#	12345678	128		True	71a6754163712443fdccafb639e7620deec10592df78ba67273c48367c462e663ede01f29451db4eb259bf98f7af3520c1fa6ae529d7f2ffa2b924cf3c688458
SHA512+Salt PLib:	GoJaguars123#	12345678	110	1000	True	6rounds=1000$12345678$X1VMqmWSWHnqBHu/b8xII9TXH0YU/KnP3cQ4o44cERqiW4gRW3osVMJZAk3obJUeXjJYL2KsRp/o3.T02mk41.
APR1MD5+Salt PLib:	GoJaguars123#	12345678	37		True	$apr1$12345678$FihZ5.XOyfYdYTkjL2b7W0
PBKDF2SHA1 Hlib:	GoJaguars123#	12345678	64	1000	True	3c6f6e2f52ef1e270b8bc648720ca7318c3d9979b742858dbce2021dd717fce2
PBKDF2SHA1 PLib:	GoJaguars123#	12345678	52	1000	True	$pbkdf2$1000$MTIzNDU2Nzg$PG9uL1LvHicLi8ZIcgynMYw9mXk
PBKDF2SHA256 PLib:	GoJaguars123#	12345678	75	1000	True	$pbkdf2-sha256$1000$MTIzNDU2Nzg$jMQu1T7poMC4CDWafPgL4q9UCBJd76Z512gvj2Vt2nM
PBKDF2SHA512 PLib:	GoJaguars123#	12345678	118	1000	True	$pbkdf2-sha512$1000$MTIzNDU2Nzg$6yI9/X6IN8sYTwyAaQaJjmPNdtr8FMpDsJ/CUBGFhlaI4bdO/lzJdnHNs47amvaiVu7s6RHlaxga/RDGaOffCQ
CTAPBKDF2SHA1 PLib:	GoJaguars123#	12345678	51	1000	True	$p5k2$3e8$MTIzNDU2Nzg=$PG9uL1LvHicLi8ZIcgynMYw9mXk=
DLITZPBKDF2SHA1 PLib:	GoJaguars123#	12345678	51	1000	True	$p5k2$3e8$12345678$jzW6x3x6LFxXF3cVfOM6gsMOmY0PxsGt

Category: Database Hashes:

Method	Password	Salt	HshLen	Rounds	Chk	Hash
MS SQL 2000:	GoJaguars123#		94		True	0x01000440881133B6A36786AD5136FC0D7B597FFC5897641114661A2C2198C60A5005A6BFA9A1E8D73E5FE18DAAD2
MS SQL 2005:	GoJaguars123#		54		True	0x01005E2BA59469BB39F12BC0EB073ACECD0D3F98F5FB642EBF7A
MySQL323:	GoJaguars123#		16		True	2eb569d21ce0efea
MySQL41:	GoJaguars123#		40		True	*210673B45102FD75CB1174ED35D10B6D2A899310
PostgresMD5Salt:	GoJaguars123#	User:12345678	32		True	md5bb48f628fefda318b4771cc3ef025666
Oracle10:	GoJaguars123#	User:12345678	16		True	ADDA187CAD42C1EA
Oracle11:	GoJaguars123#	12345678	62		True	S:D3887666DBA3AF691AFBEB95958E86D95EAC48047C18DBC04CA4090E4D7D

Category: Other Hashes:

Method	Password	Salt	HshLen	Rounds	Chk	Hash
Argon2 PLib	GoJaguars123#	12345678	87	10	True	$argon2id$v=19$m=65536,t=10,p=4$MTIzNDU2Nzg$sNZloL4rcI002e2bnYruB9xmyXSoaKSyNZmcaBAwLNw
scrypt PLib	GoJaguars123#	12345678	77	10	True	$scrypt$ln=10,r=8,p=1$MTIzNDU2Nzg$XcfwyOf8nK8uH1hfPUyBnY8wW57VJYJJw+7rcVDWtOw

Meta data extraction

- View / change meta data information of various file types

```
4. View/Change Image Info
5. View Sound File Info
6. Send Meta Data Info of file in Directory to TXT file
7. Send Meta Data Info of file in Directory to CSV file
8. Delete Meta Data Info of all files in Directory
9. Return to Main Menu
Enter integer value of Menu choice (between 1 and 9):4

Choice is: View/Change Image Info

Enter Image File name:
```

Vulnerability assessment

- Retrieves real-time vulnerability data from CISA KEV and CVE databases

```
CISA Data Menu
--------------
1. Get Known Exploited Vulnerabilities (KEV) Data
2. Get Common Exploited Vulnerailities (CVE) Data
3. Get CISA Advisory Data
4. Get Common Weakness Enumeration Data
5. Return to Main Menu
Enter integer value of Menu choice (between 1 and 5):1

Choice is: Get Known Exploited Vulnerabilities (KEV) Data


Get Known Exploited Vulnerabilities (KEVs) Data Menu
---------------------------------------------------
1. Get and Display Number of Known Exploited Vulnerabilities (KEVs)
2. Display Number of Known Exploited Vulnerabilities (KEVs)
3. Search Known Exploited Vulnerabilities (KEVs) by CVE-ID
4. Search Known Exploited Vulnerabilities (KEVs) by Vendor
5. Search Known Exploited Vulnerabilities (KEVs) by Date Added
6. Return to CISA Data Menu
Enter integer value of Menu choice (between 1 and 6):
```

Vulnerability assessment

- Retrieves real-time vulnerability data from CISA KEV and CVE databases

```
Process Common Vulnerabilities and Exploits (CVEs) Menu
-------------------------------------------------------
1. Download CVEs from CISA and extract from Zipfile
2. Read and Process Common Vulnerabilities and Exploits (CVEs) Concurrent
3. Read and Process Common Vulnerabilities and Exploits (CVEs) Linear
4. Display Number of Common Vulnerabilities and Exploits (CVEs)
5. Search Common Vulnerabilities and Exposures (CVEs) List by CVE-ID
6. Search Common Vulnerabilities and Exposures (CVEs) List by Vendor
7. Search Common Vulnerabilities and Exposures (CVEs) List by Product name
8. Write CVEID, Vendor, Product, Version, URL to JSON file
9. Return to CISA Data Menu
Enter integer value of Menu choice (between 1 and 9):
```

Large file handling

- Allow users to process very large log files
- Search log files for specific entries
- Format search results for post processing / reporting

```
File Scanning Menu
------------------
1. Read File to Scan
2. View File to Scan
3. Read Words to Search
4. View Words to Search
5. Scan File for Words
6. View Scan Results
7. Save Scan Results
8. Clear Scan File, Words, Results
```

Accessing system information

- View system details
- Can vary based on system types

```
System Info Menu
----------------
1. View Process Info
2. View Disk Info
3. View Network Info
4. View Users Info
5. View System Utilization
6. Refresh System Info
7. Return to Forensics Tools Menu
Enter integer value of Menu choice (between 1 and 7):
```

Experiments

Examples of some experiments conducted

- ***Port scanning:*** conducted port scans of several thousand ip number / port number combinations using various techniques to study effect of scan types, timeouts, etc
- ***File Checks:*** Conduct file integrity checks for very large number of files in several directory (millions) and note time to conduct tests and logging of duplicate files. Post processing of duplicate files
- ***Access Meta data information*** of file types across many directories and track to categorize files into meta data groups (camera, time, geo-location, etc)
- ***Password cracking:*** Create password hashes using various techniques and attempt password identification against well known password databases
- ***Access current CISA data*** to assess if system has vulnerable software installed

Results and Benefits

- Students involved in problem solving discussions on design variations, use of various packages, implementation variations, experiment setup, etc
- Hands-on understanding of cybersecurity principles, scripting, and real-world security challenges.
- Enhances critical thinking and analytical thinking for solving problems involving cybersecurity concepts
- Students build successful applications that can be enhanced for usage in various environments
  - ***Example:*** A photographer was able to see the meta data stored from various pictures (camera types, etc) and glean additional information about photographer profile
  - ***Example:*** Working on meta data of images/files from abandoned devices (laptops, cameras, cellphones, etc), a person attempted to create social network connection profiles and try to search for similar profile connections on popular social networks to establish identity.

Conclusion and future work

- This presentation outlines the development of a practical cybersecurity toolkit by undergraduate students, that integrates key functionalities for security analysis.
- Future work will focus on enhancing the toolkit with a graphical interface, AI-driven threat detection, and expanded forensic capabilities.

Acknowledgement

- This work was completed as part of a consortium grant entitled **“Military City – USA”**. The grant is funded by the U.S. Department of Defense (DOD) under the National Defense Education Program (NDEP) to support the advancement of Science, Technology, Engineering, and Mathematics (STEM) pathways for community college students in San Antonio. Members of the consortium include colleges in the Alamo System, Texas A&M University – San Antonio, and University of Texas – San Antonio.