

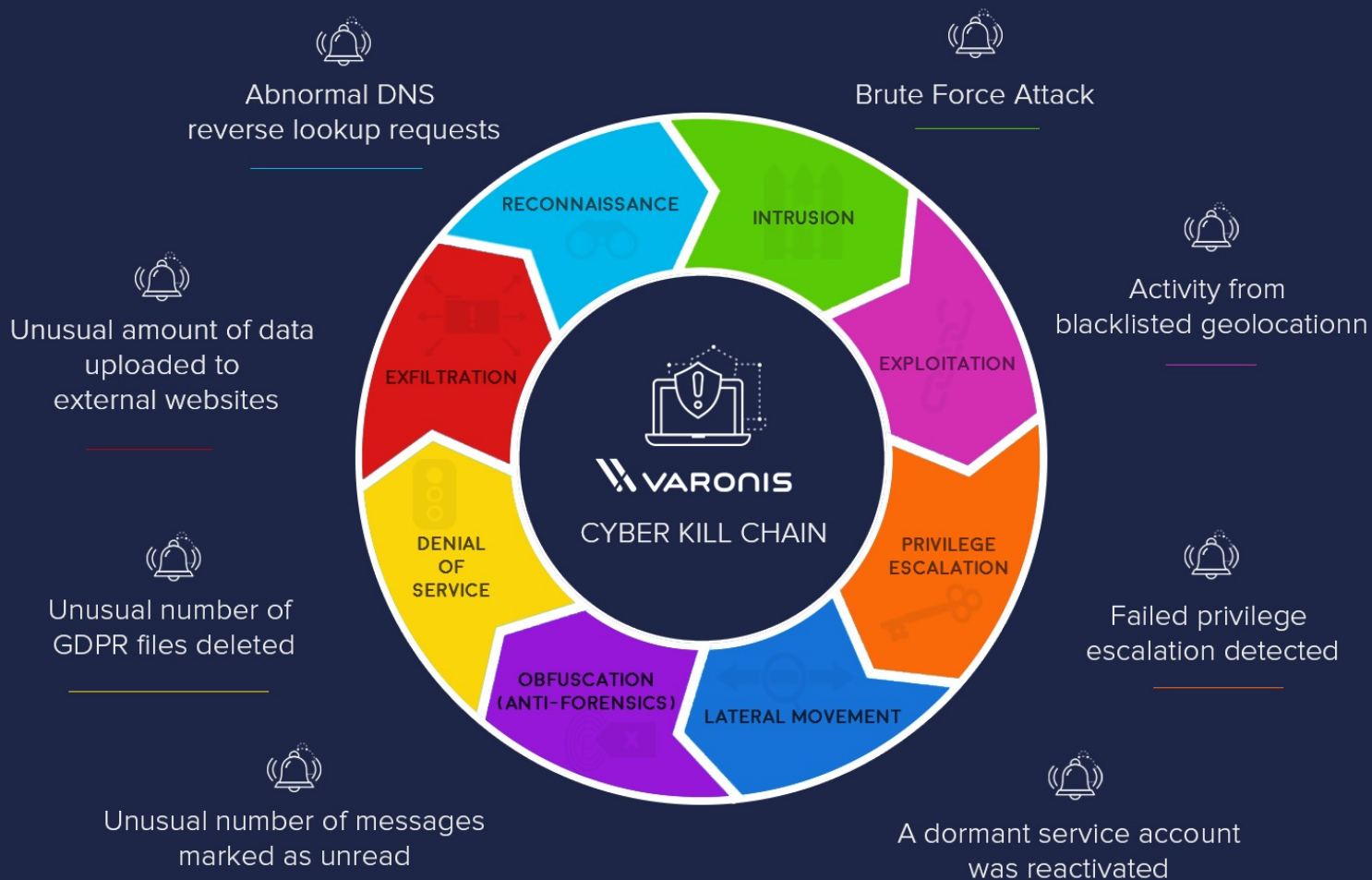


Computer Network Security

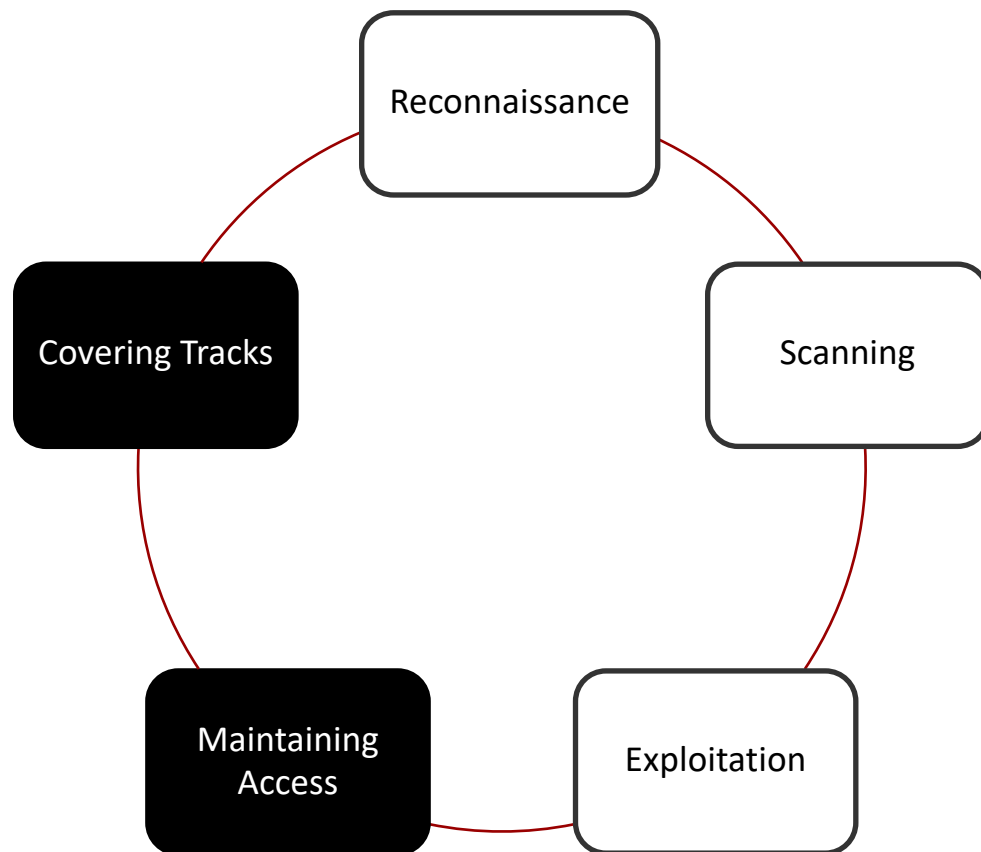
COMP 178 | Spring 2022 | University of the Pacific | Jeff Shafer

Penetration Testing: Reconnaissance

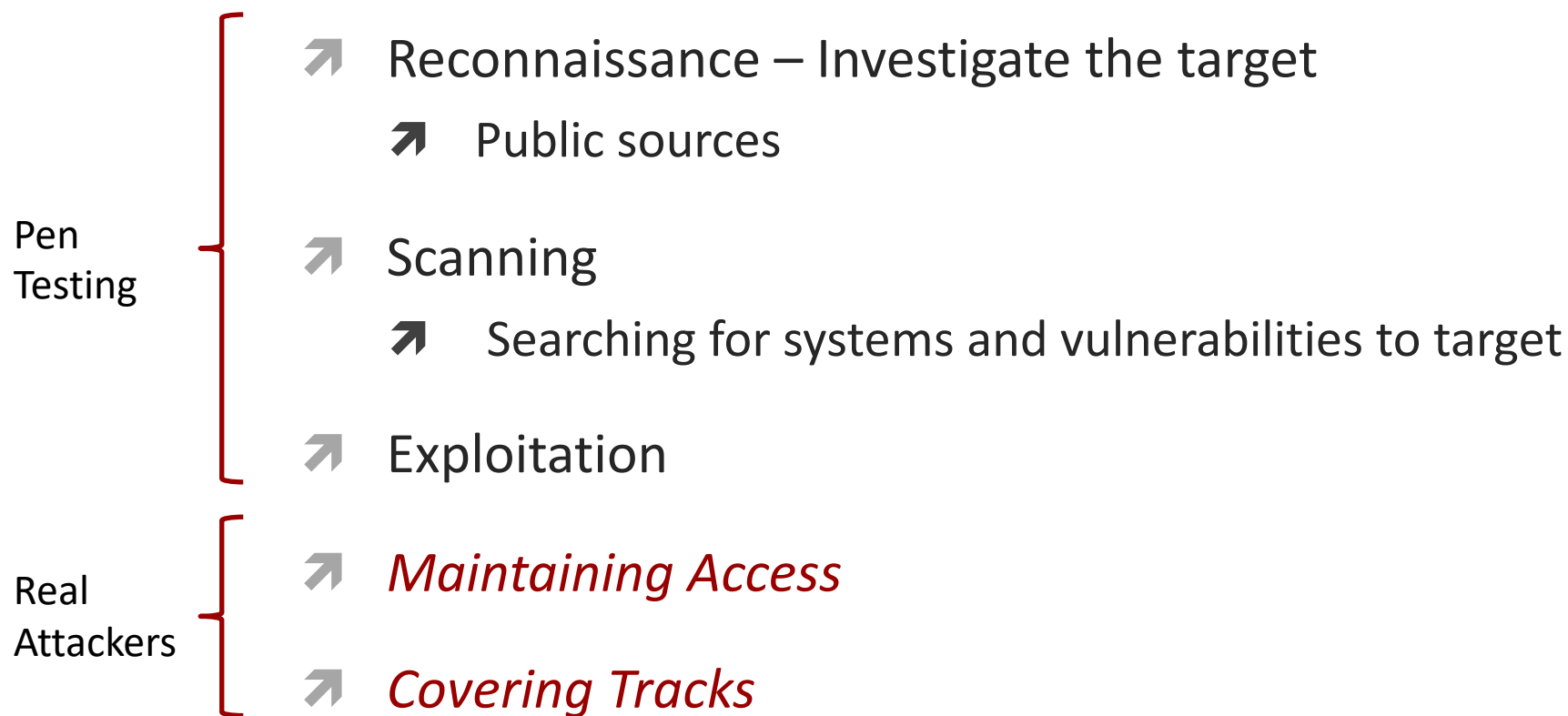
Stages of "Cyber Kill Chain"



Stages of an Attack



Stages of an ~~Attack~~ Pen Test



Reconnaissance Stage

- Spend a day gathering information about the target from **public sources**
 - Not *necessarily* technical! (tech-lite?)
 - Not doing scans of their network (that comes next)
 - Google, LinkedIn, myriad public databases

- Put yourself in the shoes of the target and better understand them
 - What business are they in?
 - Where are their facilities located?
 - Who works there? (names, titles, emails, ...)
 - Bonus points if you can find names & emails of executives and IT administrators – Would be *very* helpful to get their logins later!
 - What kind of technology do they claim to use?

Reconnaissance Stage

➤ Benefits

- Improves your chances of success during the rest of the pen test
- Reporting documents (of threats, risks, ...) can be tailored to the customer instead of being generic

Resources

- **Penetration Testing Framework**
- Detailed checklist of common pen-test areas of interest
 - Reconnaissance
 - Discovery & Probing (Scanning)
 - Password Cracking
 - Network Testing
 - Penetration
 - Physical Security

<http://vulnerabilityassessment.co.uk/Penetration%20Test.html>

Resources

- **Open-Source Intelligence (OSINT)**
 - Gather information from free databases and resources
 - *Deeper than just “Google it!”*

- Applied during pentest reconnaissance stage
 - Take a small piece of information and leverage it to learn more
 - Example: Given a domain name, what can I learn?
 - Whois records to check ownership
 - Enumerate (guess/lookup) subdomains
 - Check for SSL certificates
 - ...

Resources

- OSINT Framework - <https://osintframework.com/>
- Skip Tracing Framework - <https://makensi.es/stf/>
 - *Last updated in 2012, so some of the tools are broken, but the ideas are still useful*

System Inventory

- You won't remember everything you discover during reconnaissance and scanning! Need an organized method to document your discoveries
- One suggestion: **System Inventory**

IP Addr	Hostname	OS	Discovery Method	Open Ports	Vulns?	Accounts & Passwords	Notes

Discovery Method

- **Discovery Method** is particularly important to document thoroughly – it makes your results *reproducible*
 - Provided by customer? (during scoping discussions)
 - Google search?
 - DNS? (zone transfers, reverse lookups, etc...)
 - Network scans? (ICMP, TCP, UDP, ...)
 - Physical access?
 - Pivoting through another host? (which host?)

Automated Documentation - Dradis

Learn about 1-click reporting – Dradis Professional Edition

dradis-pro.dev/nodes/9

Learn about 1-click reporting

Upload output from tool | Export results | Change project | Configuration

All issues | Methodologies

Nodes

scope

hosts

- 10.0.0.1
- 10.0.0.2

+ Add a new node

Nodes / hosts / 10.0.0.1

Notes +

Did you know...?

Issues +

- Out-of-date Apache
- SSLv2 is enabled
- Directory listings enabled

Attachments

200x200.png
32.9 KB

Drop zone

Evidence for this instance [edit](#) [remove](#)

Port
tcp/443

Output
It was possible to establish an SSLv2 connection with the server.

```

---
SSL handshake has read 2048 bytes and written 364 bytes
---
SSL-Session:
  Protocol   : SSLv2
  Cipher     : DES-CBC3-MD5
  Session-ID: 1D3500005F08024F660E5C2E6D5134E2
  Session-ID-ctx:
  Master-Key: AAC0F75EF7E7AFC062A1FE79CF4401F3CEDDE85CF892C585
  Key-Arg    : C66525AFEB39F353
  Start Time: 1390379241
  Timeout    : 300 (sec)
  Verify return code: 21 (unable to verify the first certificate)
---

```

Dradis Professional v1.10.0.pre

Automated Documentation - MagicTree

The screenshot shows the MagicTree application interface. On the left is a 'Tree View' showing a hierarchy: 'magictree' (33) containing 'testdata' (33) with sub-items 'netblock 192.168.1.0/24', 'netblock 192.168.2.0/24', 'host 192.168.1.1' (9), 'host 192.168.1.100' (13), 'host 192.168.1.101' (3), 'host 192.168.1.104' (3), and 'host 192.168.1.106' (3); and 'repo' and 'tasks'.

The main area is split into 'Table View' and 'Tasks'. The 'Table View' shows a query: 'Query/Method not saved in repository'. It contains a table with columns: Title, Expression, Leaf, and Hidden.

Title	Expression	Leaf	Hidden
host	//host	<input type="checkbox"/>	<input type="checkbox"/>
protocol	ipproto	<input type="checkbox"/>	<input type="checkbox"/>
port	port[state="open"]	<input type="checkbox"/>	<input type="checkbox"/>

Below the table are buttons: Run, Stop, < Prev, Next >, Clear, and Save.

Below that, it says 'Found 12 row(s)' with buttons 'Copy to Clipboard' and 'Clear'. The 'Table cell click action' is set to 'none' (selected), with options for 'select' and 'filter'.

host	protocol	port
192.168.1.1	tcp	21
192.168.1.1	tcp	22
192.168.1.1	tcp	23
192.168.1.1	tcp	80
192.168.1.100	tcp	21
192.168.1.100	tcp	23

At the bottom of the table view, there are input fields and buttons: 'Input' (12 rows, 3 field(s): host, protocol, port), 'Environment' (radio button), 'TabSep in \$in file' (radio button), 'Run', 'Command' (sudo nmap -sS -sV -O -oX \$out.xml -iL \$in), 'Save', 'User@Host', and 'Push SSH key'.

The 'Message Log' at the bottom shows:


```
20:04.31 Using mtdir '/home/alla/.magictree'
20:04.31 Initializing MagicTree Beta Two, rev 1243
20:04.31 Successfully imported XML data
```

- Google is your friend!
 - All kinds of data is intentionally (or unintentionally) placed online
 - Search for interesting files?
 - `site:example.com filetype:.pdf`
 - Search for Unix passwd file?
 - `site:example.com intitle:index.of passwd`
 - (Password is hashed but usernames are present)

Google Dorks

- Google Hacking Database, aka “Google Dorks”
 - Simple Google queries to pull up potentially vulnerable systems or interesting information

- Categories
 - Files Containing Juicy Info
 - Sensitive Directories
 - Vulnerable Servers
 - Network or Vulnerability Data
 -

<https://www.exploit-db.com/google-hacking-database>

Google Dorks

- Example – Pfsense login pages
 - Pfsense is a firewall – do you have an exploit handy and are looking for targets?
 - `intitle:"Pfsense - Login"`
 - <https://www.exploit-db.com/ghdb/5671>

- Example – Cacti Monitoring
 - Cacti is a network monitor, and might show servers that the enterprise thinks are important enough to monitor
 - `intitle:"Cacti" AND inurl:"/monitor/monitor.php"`
 - <https://www.exploit-db.com/ghdb/5600>

<https://www.exploit-db.com/google-hacking-database>



Class Presentations



Class Presentations

- **Topic:** Anything related to computer security, network security, etc...
- **Scope: 15 minutes of technical content w/slides**
- **Timeline**
 - **Proposal due – Tue Feb 8th**
 - Video & slides due – Tue Mar 1st
 - Peer Reviews due (3) – March 11th

Class Presentation

➤ Ideas for inspiration?

<https://cyberlab.pacific.edu/resources/security-and-privacy-news>