Post Exploitation & Data Exfiltration

ohdae@beacon:~# whoami

Bindshell Labs

Intersect Framework

metasploit®

python™

What is Post Exploitation?

- Everything that you do after your initial exploitation and entry onto a target

- Determine value of compromised system
  - what do they have?
  - what do I want?

- Gather desired information
  - passwords, identity theft, documents, exfil...

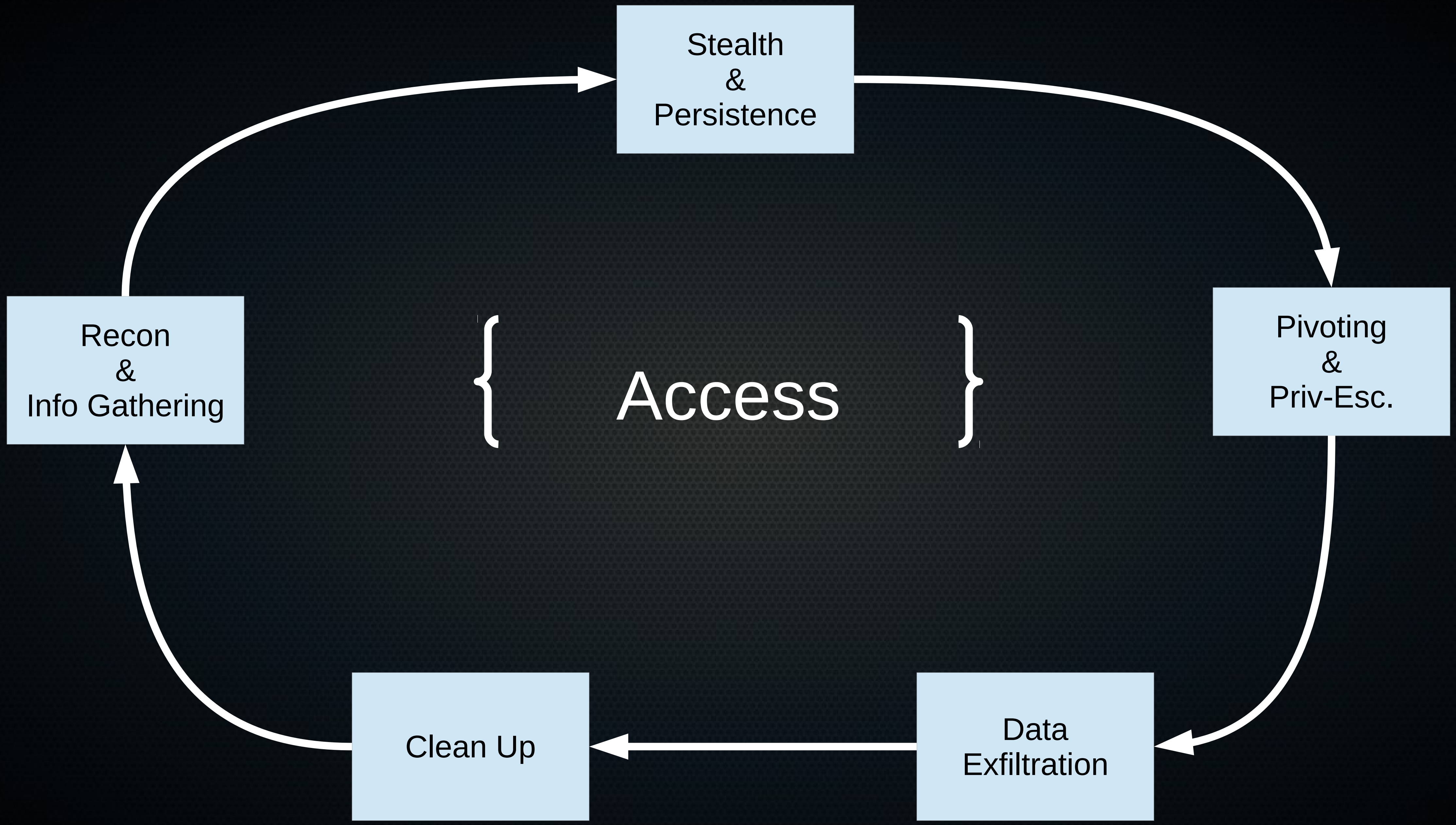
- Maintain access
  - backdoors, legitimate access, etc.

Problems

- Very little standardization
- Lack of automated tools
- Many testers don't do enough
- 'DA / Root is all that matters' mentality

Why Post Exploitation?

- This is the stuff that matters!
- Shows realistic impact of a breach
- Provides best value to your clients
- Companies sell products.
- This is what makes them money.



Stealth & Persistence

Pivoting & Priv-Esc.

Data Exfiltration

Clean Up

Recon & Info Gathering

{ Access }

Information Gathering

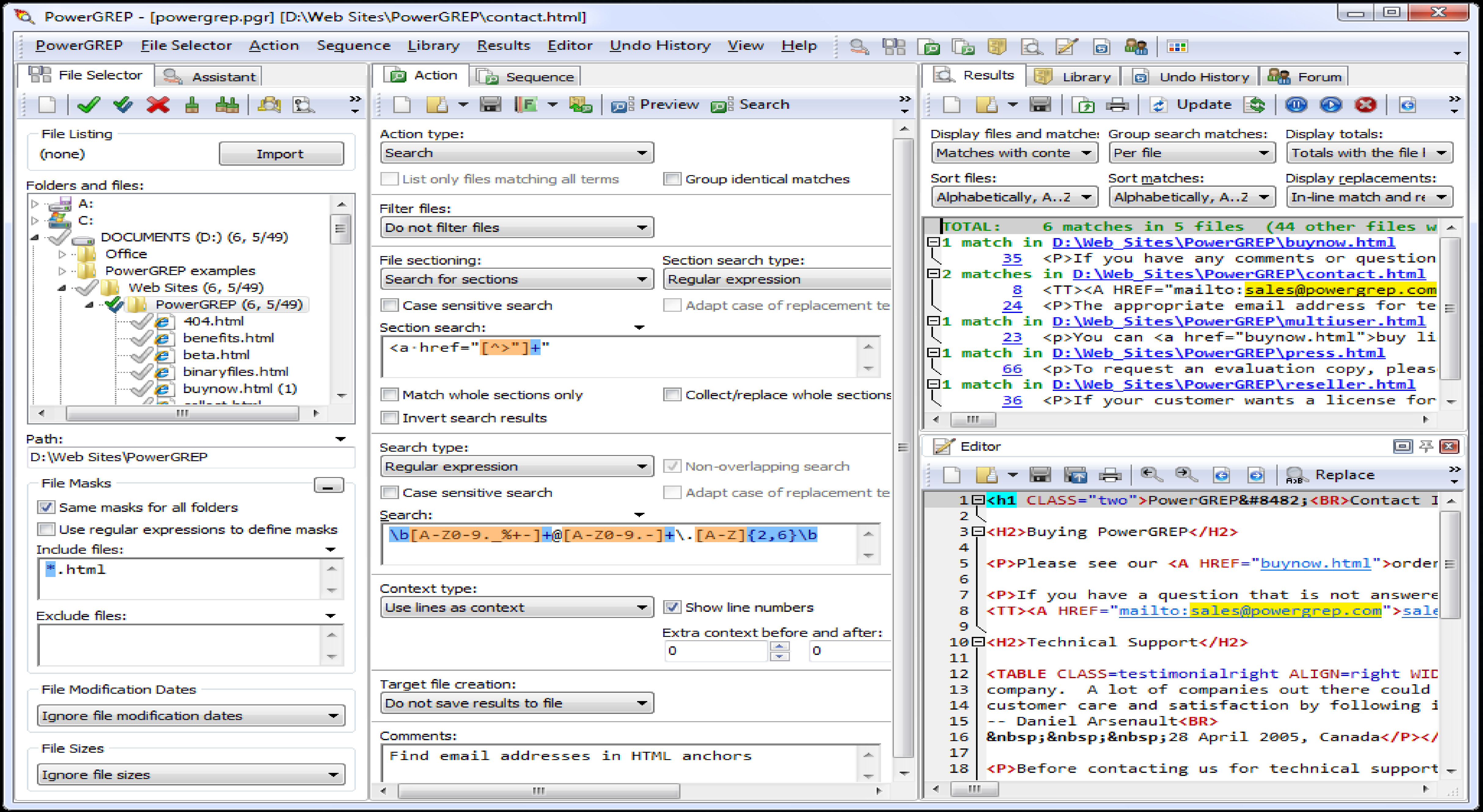
- **Have a plan ahead of time!**
- **What does this system have that I want?**
- **Authentication & Access levels**
- **Document & Log as you go!**

- **User Identities**
  passwords, keys, etc.
- **Network Information**
  services, open ports, firewall rules, egress filters, internal mapping...
- **System Information**
  distro, patch level, back-ups, file systems, devices, etc.
- **Files, Documents, Data**
  config files, office documents, code repos, client lists, financial, etc.

All Data Is Not Created Equal

- You do not need **everything** from **every** system!
- Prioritize what data is most important
- What is most valuable to the client company?

- **Valuable to the client means valuable to you!**
- Smash and grab is fun but not very efficient!
- Use central location for data you are taking



File Listing
(none)

Folders and files:
- A:
- C:
- DOCUMENTS (D:) (6, 5/49)
  - Office
  - PowerGREP examples
  - Web Sites (6, 5/49)
    - PowerGREP (6, 5/49)
      - 404.html
      - benefits.html
      - beta.html
      - binaryfiles.html
      - buynow.html (1)

Path: D:\Web Sites\PowerGREP

File Masks
☑ Same masks for all folders
☐ Use regular expressions to define masks

Include files:
```
*.html
```

Exclude files:

File Modification Dates: Ignore file modification dates

File Sizes: Ignore file sizes

Action type: Search

☐ List only files matching all terms
☐ Group identical matches

Filter files: Do not filter files

File sectioning: Search for sections

Section search type: Regular expression

☐ Case sensitive search
☐ Adapt case of replacement te

Section search:
```
<a·href="[^>"]+"
```

☐ Match whole sections only
☐ Collect/replace whole sections
☐ Invert search results

Search type: Regular expression

☑ Non-overlapping search
☐ Case sensitive search
☐ Adapt case of replacement te

Search:
```
\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}\b
```

Context type: Use lines as context

☑ Show line numbers

Extra context before and after: 0 0

Target file creation: Do not save results to file

Comments:
```
Find email addresses in HTML anchors
```

Display files and matche: Matches with conte

Group search matches: Per file

Display totals: Totals with the file l

Sort files: Alphabetically, A..Z

Sort matches: Alphabetically, A..Z

Display replacements: In-line match and re

```
TOTAL:     6 matches in 5 files  (44 other files w
1 match in D:\Web_Sites\PowerGREP\buynow.html
       35  <P>If you have any comments or question
2 matches in D:\Web_Sites\PowerGREP\contact.html
        8  <TT><A HREF="mailto:sales@powergrep.com
       24  <P>The appropriate email address for te
1 match in D:\Web_Sites\PowerGREP\multiuser.html
       23  <p>You can <a href="buynow.html">buy li
1 match in D:\Web_Sites\PowerGREP\press.html
       66  <p>To request an evaluation copy, pleas
1 match in D:\Web_Sites\PowerGREP\reseller.html
       36  <P>If your customer wants a license for
```

```
<h1 CLASS="two">PowerGREP&#8482;<BR>Contact I

<H2>Buying PowerGREP</H2>

<P>Please see our <A HREF="buynow.html">order

<P>If you have a question that is not answere
<TT><A HREF="mailto:sales@powergrep.com">sale

<H2>Technical Support</H2>

<TABLE CLASS=testimonialright ALIGN=right WID
company.  A lot of companies out there could
customer care and satisfaction by following i
-- Daniel Arsenault<BR>
&nbsp;&nbsp;&nbsp;28 April 2005, Canada</P></

<P>Before contacting us for technical support
```

Persistence

- The longer you have access, the more 'damage' can be done
- Some exploits and attacks are a one-shot deal (i.e., SE attacks)
- Real attackers plan on hanging out for a long time

- Legitimate access is always the best choice
- Systems get patched, updated, etc.
- Lost shells

Persistence Methods

- **SSH**
  Steal keys, insert your own, backdoor service
- **Start-up Service**
  xinetd, initd, Windows registry..
- **Vulnerable Apps**
  Put vulns into existing services or applications
- **Time-Based**
  cron, AT, custom scripts, C&C style, Matahari script
- **Custom**
  Backdoor acct creations, user profiles, etc.
- **Multiple Methods**
  Always have another way in!

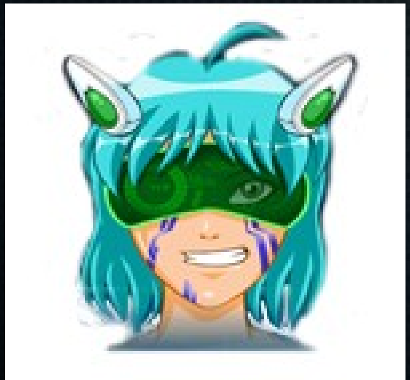
Data Exfiltration

- Store collected data in centralized location
- Encryption during transit
- Hide in plain sight
- Legitimate transfer don't stand out
- Persistent + Exfiltration = WIN

- Only take what you need
- Although, credentials are always good to steal ;-)
- **Methods**
  Netcat
  Meterpreter
  Socat
  Intersect
  SFTP
  Webserver
  DNS
  ICMP
  Email
  Tunneling

Automation...or the lack of it..

- Post-exploitation is time consuming...if you do a good job
- Few tools exist to provide automation, mainly Meterpreter
- Automate the information gathering, recon, network mapping, exfilration...
- Keep logs of what tasks the automation performs!



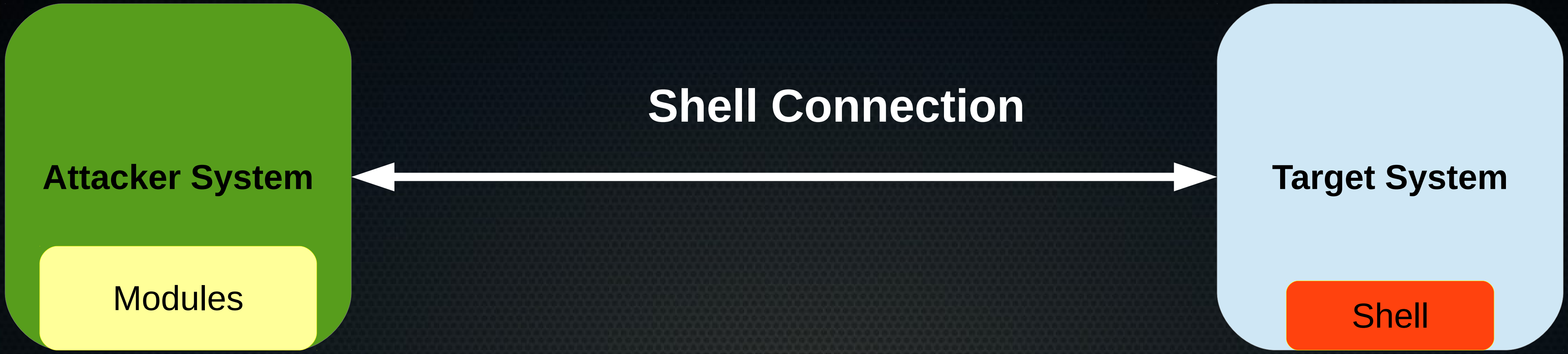
metasploit

Weeveley

Intersect

Intersect Framework

- Written completely in Python
- Linux support only (for the time being)
- Client-Server or Local scripts
- Modules to automate variety of post-exploitation tasks
  - Information Gathering
  - Privilege Escalation
  - Persistence
  - Logging of tasks
  - Network mapping
  - Reverse & Bind shells



- Target system runs small shell script
- Modules stored on attacker system
- Sent to target on demand
- Modules stored, read and executed from shell script memory
- Information is gathered & piped back to attacker

Too long, didn't read

- Create and implement a plan of action
- Information gathering is key
- The value of a penetration test is in the data, not the amount of root shells you get!
- With 'legitimate' persistent access and exfiltration, you can stay inside a system forever
- We have hundreds of automated tools for attacks, not many for post-exploitation.
- Document as you go, it's much easier!

Acknowledgements

- BeaCon & BeanSec crew
- Bindshell Labs IRC
- #metasploit freenoders
- All of you people for listening!

Bindshell.it.cx **@bindshell_** **github.com/ohdae**