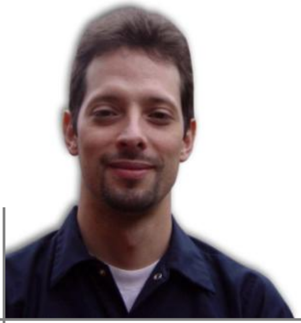
State of the Art Security from an Attacker's Viewpoint

Ivan Arce · Gerardo Richarte

Core Security Technologies
46 Farnsworth St
Boston, MA 02210
Ph: (617) 399-6980
www.coresecurity.com

We work for **Core Security Technologies** | http://www.coresecurity.com

Who we are?



Ivan Arce

CTO

Gerardo Richarte

Lead Exploit developer

Associate Sr. Consultant

What do we do?

CORE IMPACT	Software that automates Penetration Testing
SCS SECURITY CONSULTING SERVICES	Penetration Testing and Software Security Auditing services

We will discuss information security from an attacker’s perspective

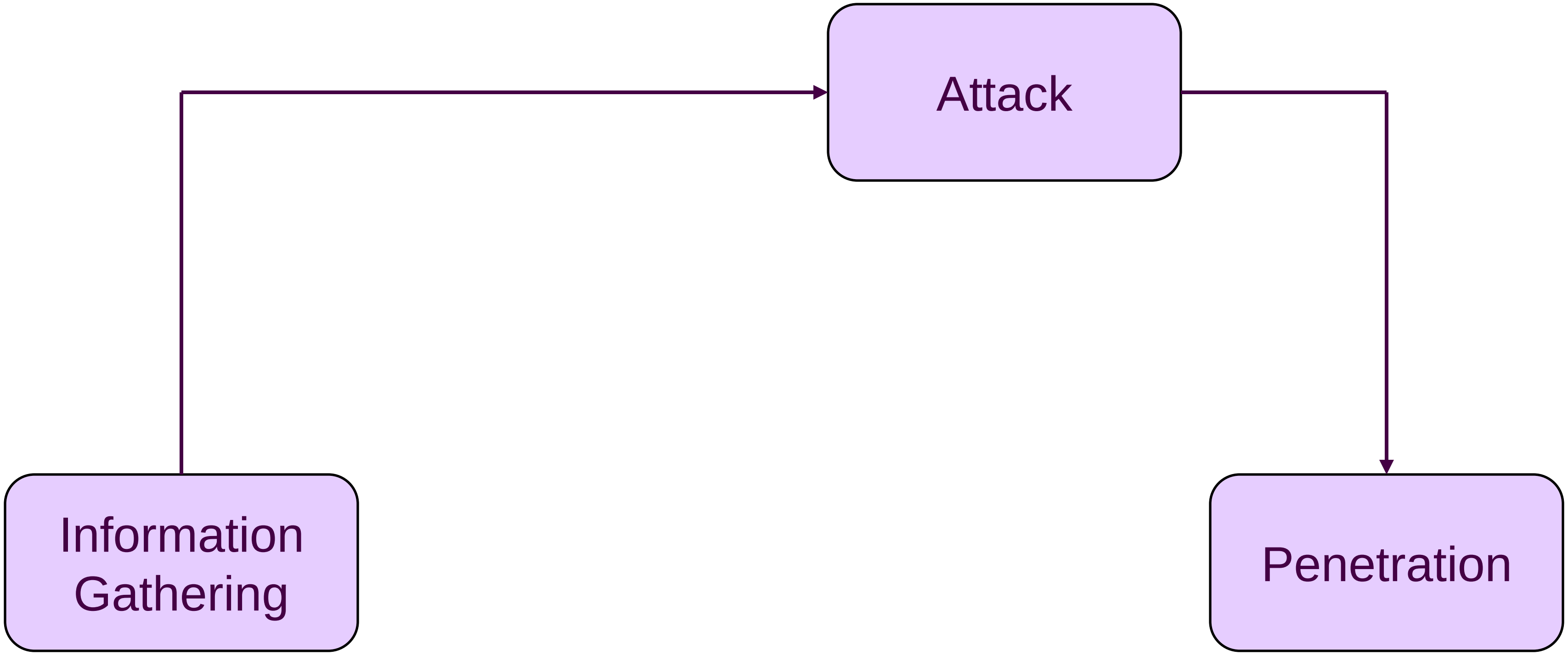
OUTLINE

- Current Attack & Penetration practices
- Privilege Escalation and Pivoting
- Other attack targets
- Attack planning and modeling

To improve our security posture we need to understand the attacker

ATTACK AND PENETRATION

Mass-rooters and ‘skript kiddies’ use the simplest attack methodology

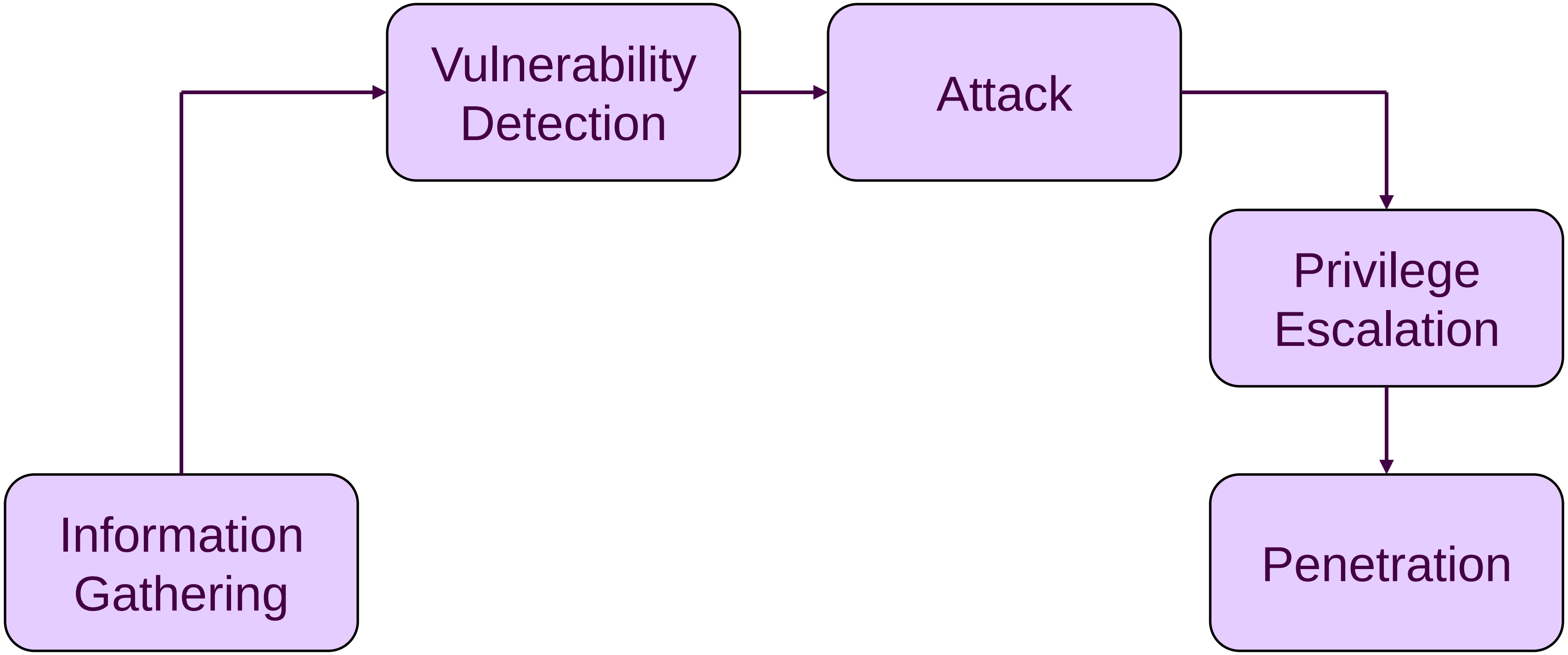


Information Gathering

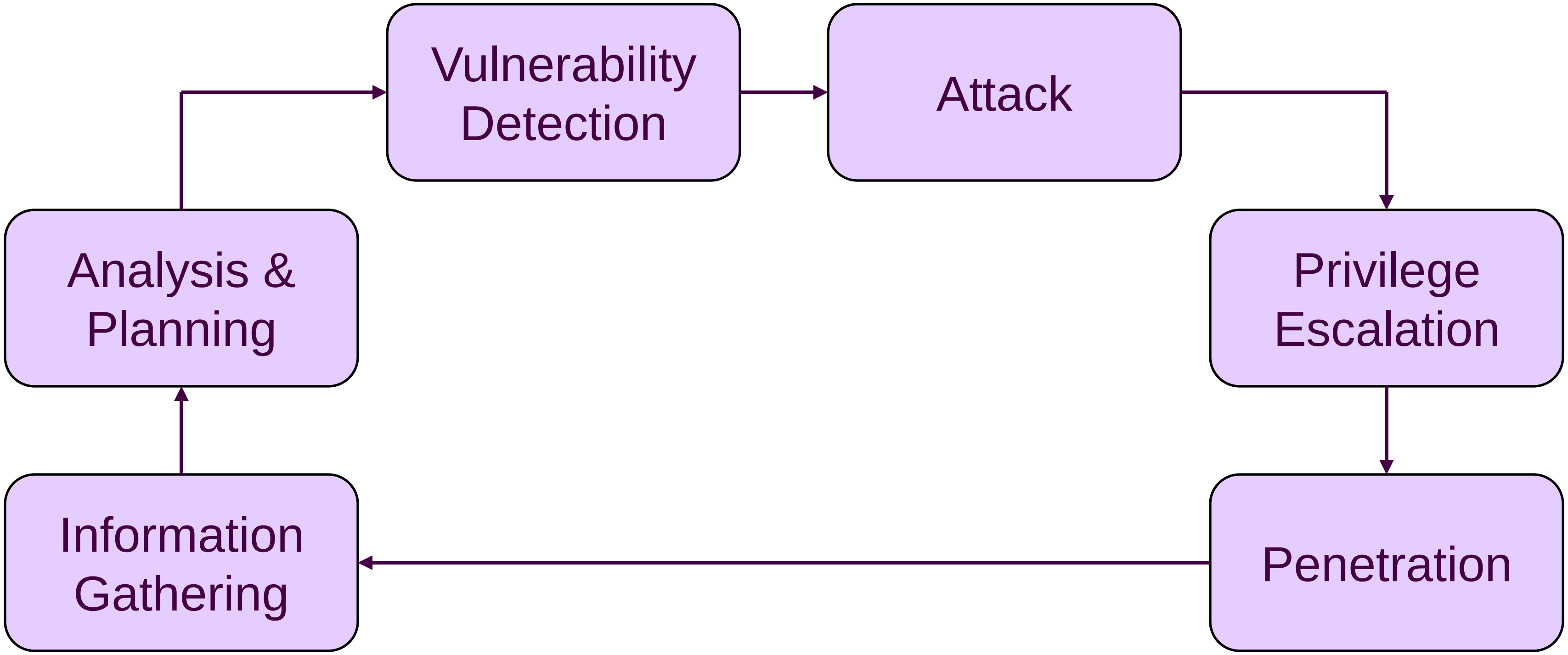
Attack

Penetration

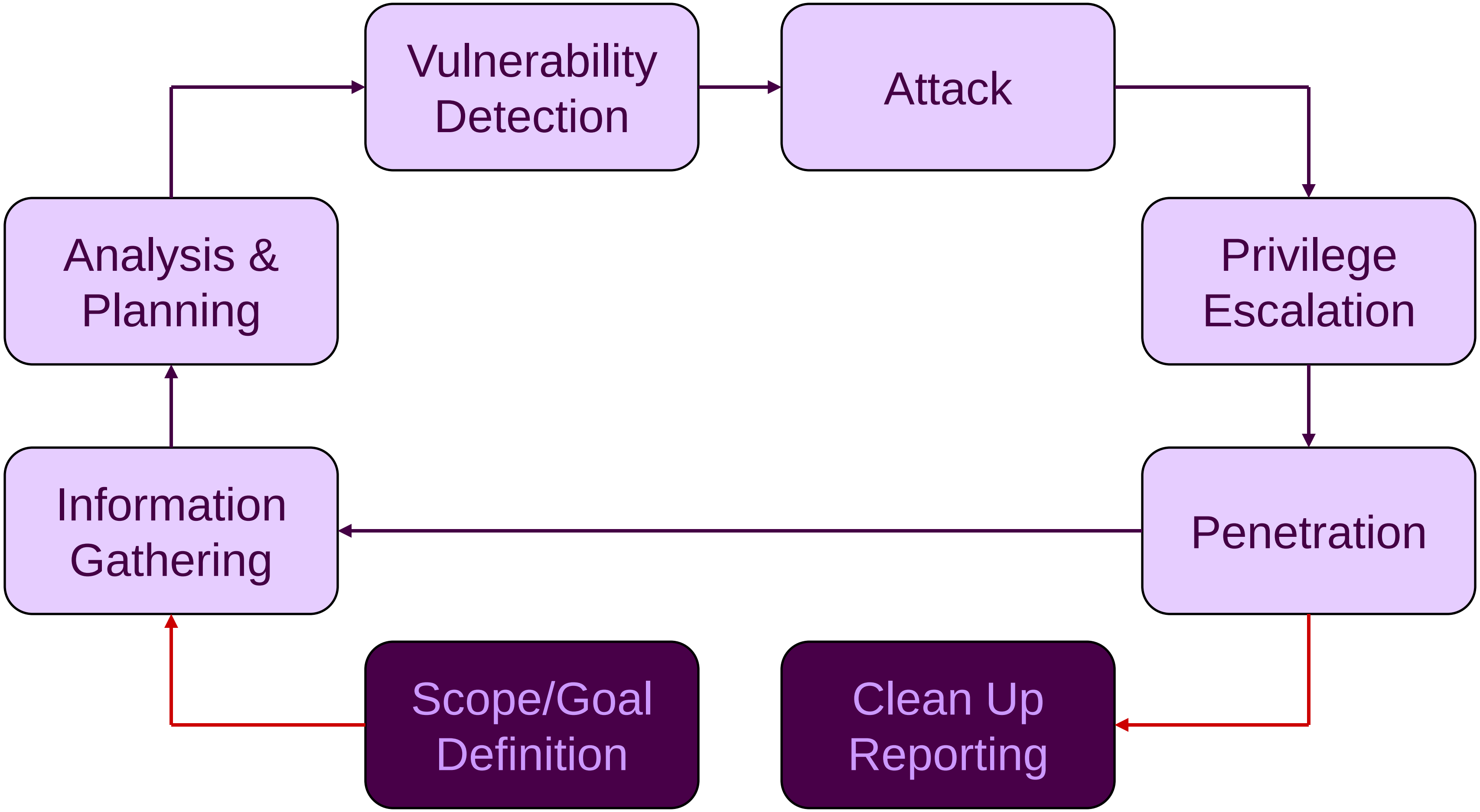
A dedicated attacker adds extra steps to increase success rate



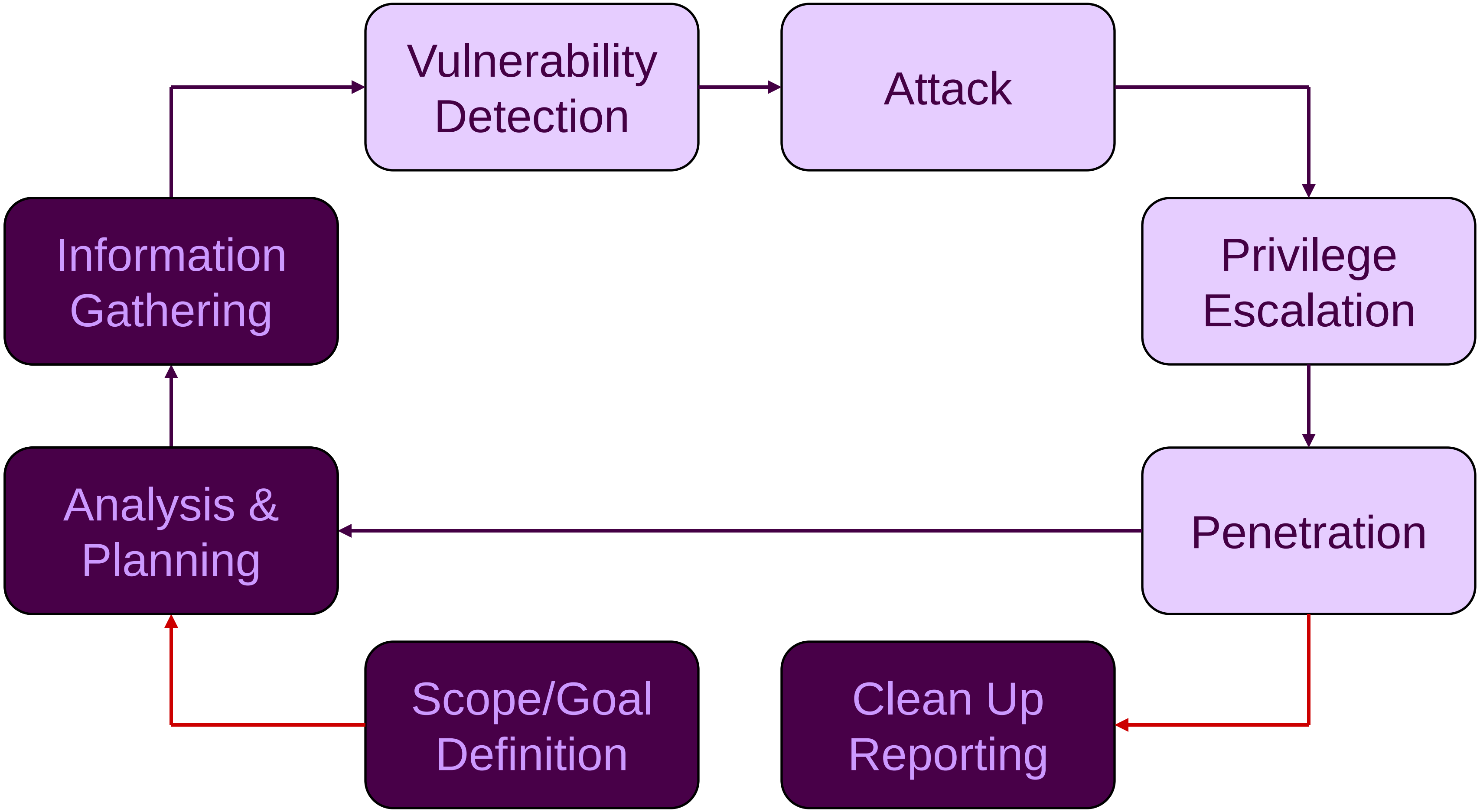
Sophisticated attackers plan ahead and go deeper



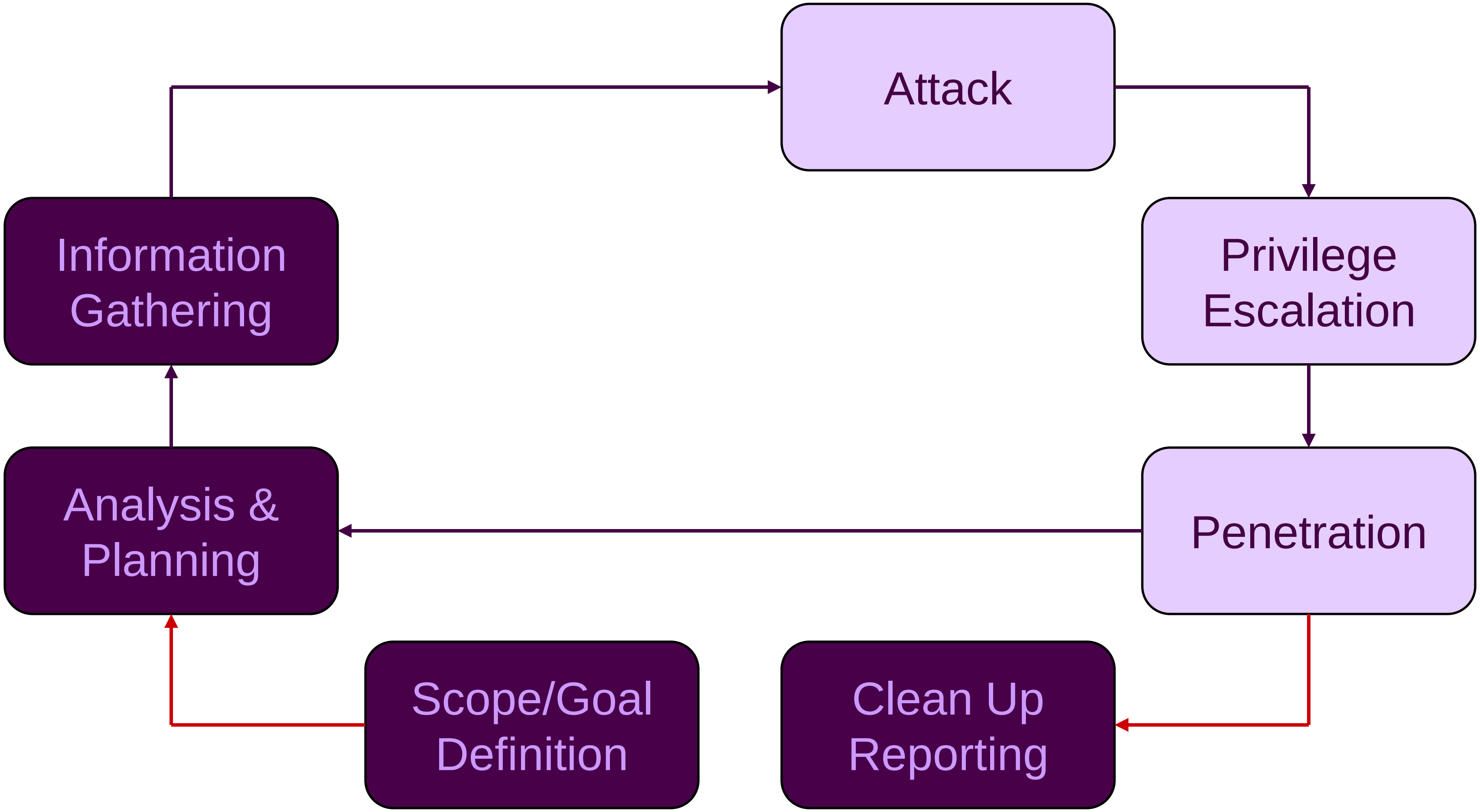
Professional penetration testers must fit in business criteria



Penetration testing efficiency can be improved with methodology



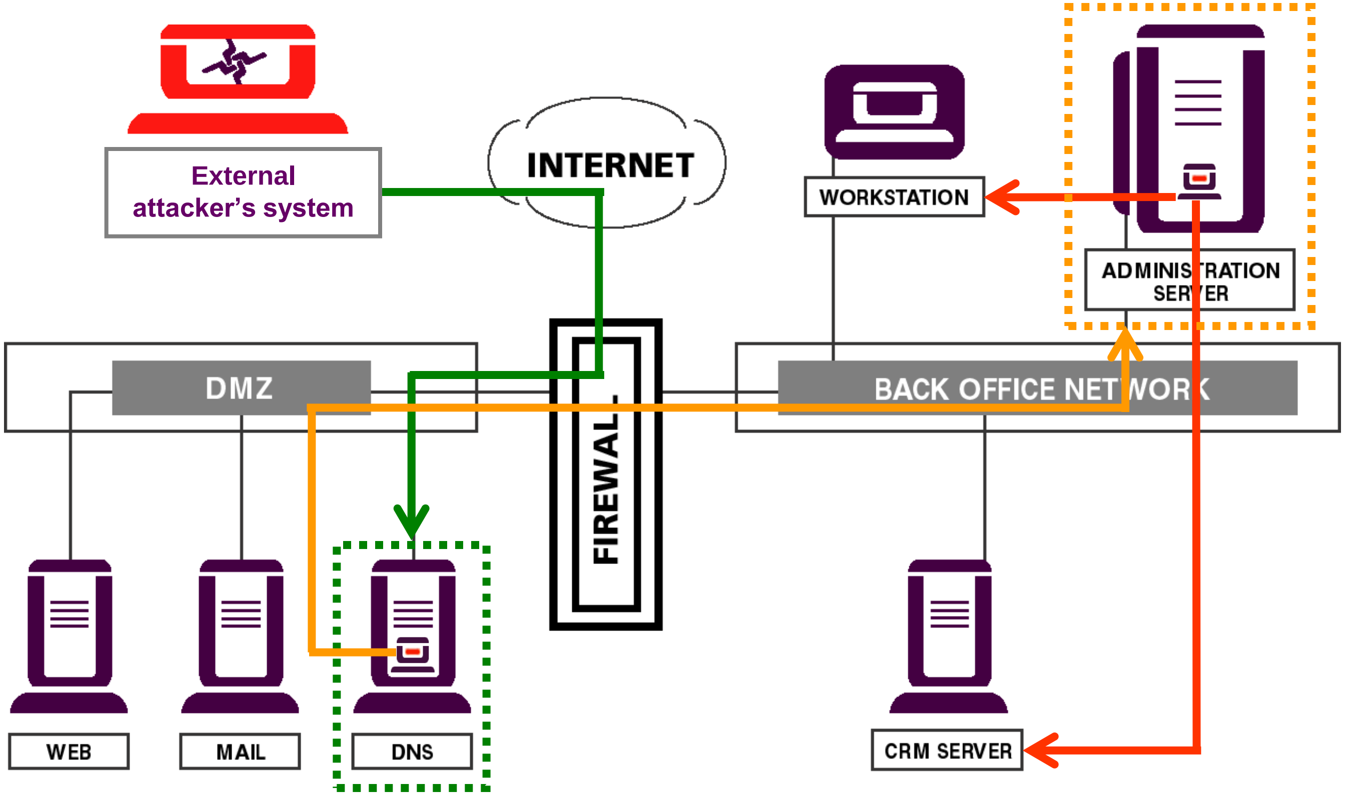
And still mimic the most basic attack scenarios…



PRIVILEGE ESCALATION AND PIVOTING

A sophisticated real-world attacker will leverage trust relationships to gain access to more valuable information assets

ANATOMY OF A REAL-WORLD ATTACK



Base camp

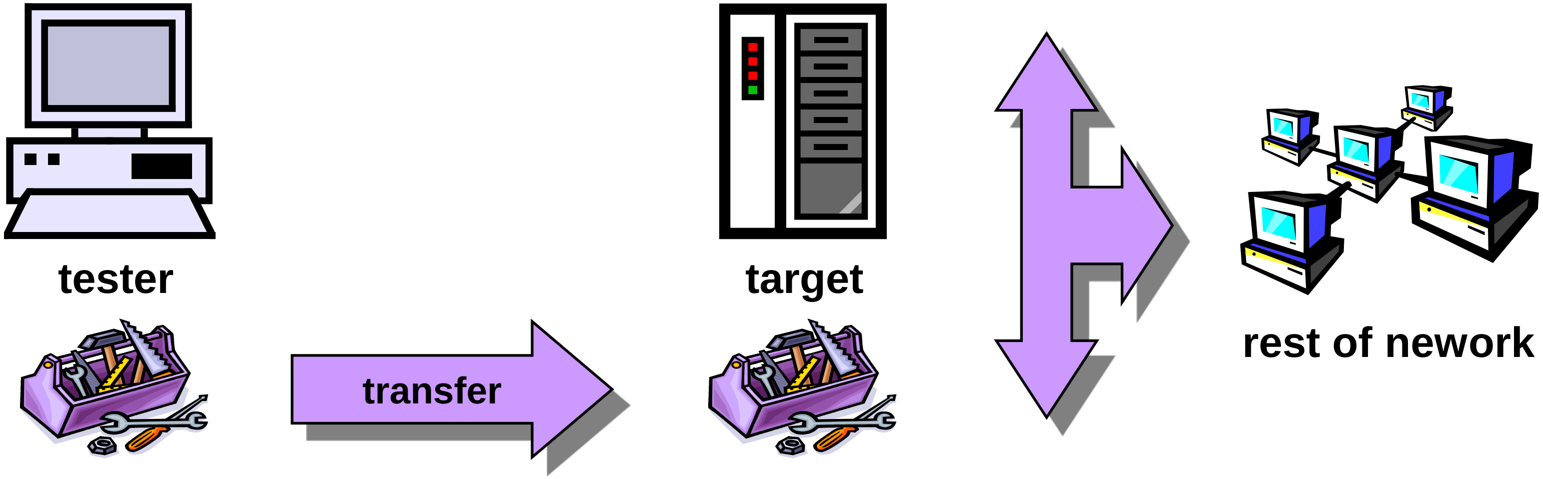
A target server is attacked and compromised

The acquired server is used as vantage point to penetrate the corporate net

Further attacks are performed as an internal user

Pivoting can be a complex and time-consuming step

- After successful attack against a target
- Use the compromised host as a vantage point (pivoting)
  - Attacker profile switch: from external to internal
  - Take advantage of the target credentials within its network
  - Exploit trust relationships
- To be able to pivot, the tester needs his tools available at the vantage point



There are several methods used to maintain access to a compromised system

COMMON PENETRATION TECHNIQUES

- Add direct shell access on a listening port

```
echo "ingreslock stream tcp nowait root /bin/sh sh -i" >>/tmp/bob ; /usr/sbin/inetd -s /tmp/bob &"
```

- Add a new account to the compromised system

```
echo "sys3:x:0:103::/:/bin/sh" >> /etc/passwd;
echo "sys3:1WXmkX74Ws8fX/MFI3.j5HKahNqIQ0:12311:0:99999:7:::" >> /etc/shadow
```

- *Use a “call home” command shell*

- *Install backdoor using existing binaries*

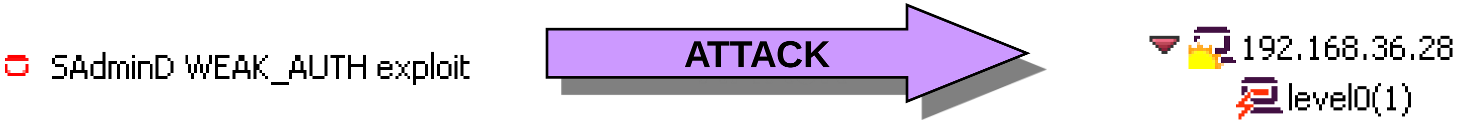
SSH daemon, telnetd, , etc.

- *Install rootkits to ensure access, establish cover channel and minimize detection*
- *Enhance attack payload (shellcode) to provide the techniques described above*

Agents provide seamless pivoting after successful exploitation

USING AGENTS AT THE BASE CAMP

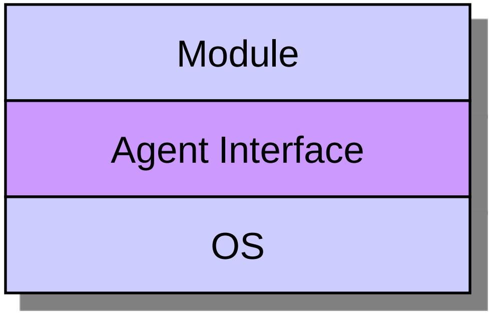
- Exploits deploy an agent on compromised systems
  - Payload is independent from exploitation specifics
  - Payload is independent from settings not related to exploitation technique
  - Payload is platform dependant
  - Suppy small agent as attack payload
    - Agent highly optimized for size (Linux agent ~80 bytes, Windows ~180bytes)
    - Agent inherits privileges of vulnerable program

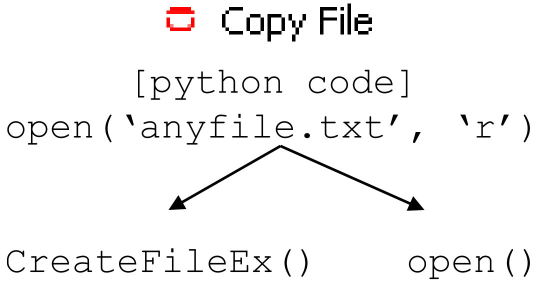


- Benefits
  - Transparent pivoting
  - “Local” privilege escalation
  - Doesn’t rely on the presence and availability of a shell
  - Easy to clean up

Agents provide platform independence

- Provides a uniform layer for interacting with the underlying system
  - Generic modules are platform independent
  - Porting the agent to different platforms effectively makes all modules available on that platform





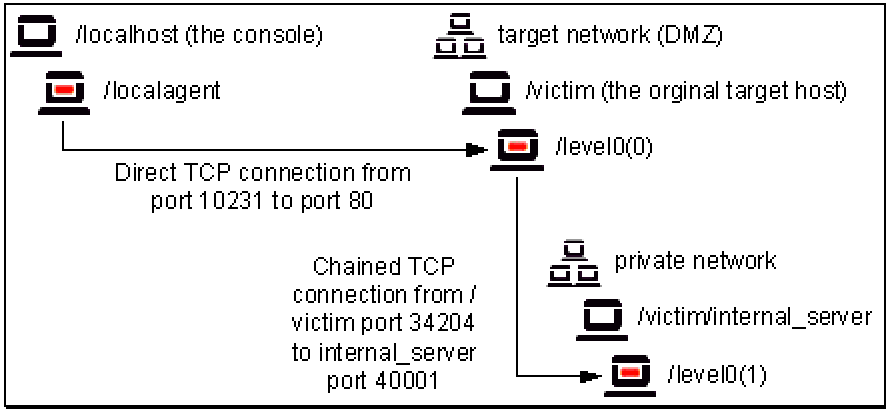
```
[python code]
open('anyfile.txt', 'r')

CreateFileEx()     open()
```

- Isolates the particular characteristics of the pivoting host platform from the module
  - Simplifies module development
  - Simplifies product use

Agents are automatically chained to assure connectivity

AGENT CHAINING

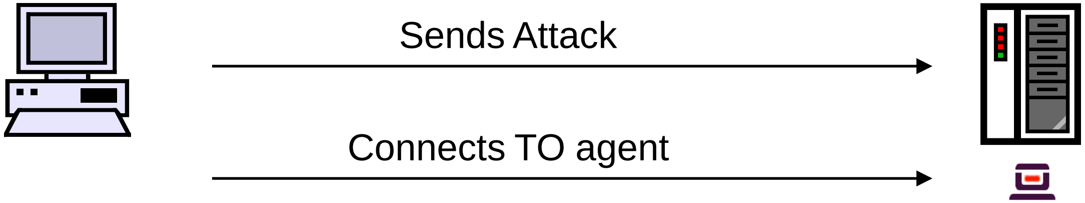

/localhost (the console)
/localagent
Direct TCP connection from port 10231 to port 80
target network (DMZ)
/victim (the orginal target host)
/level0(0)
Chained TCP connection from /victim port 34204 to internal_server port 40001
private network
/victim/internal_server
/level0(1)

- Automatic: agents are chained to the current source agent (*implicit chaining*)
- Enables the tester to communicate with agents deep into the target network

Multiple agent connection methods aid in providing connectivity in different network environments

AGENT CONNECTION METHODS

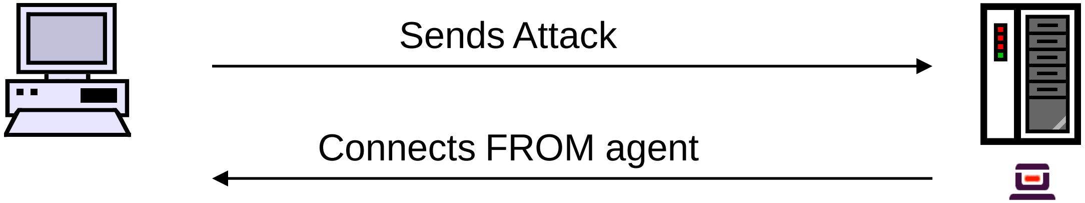
- Connect to target



Sends Attack

Connects TO agent

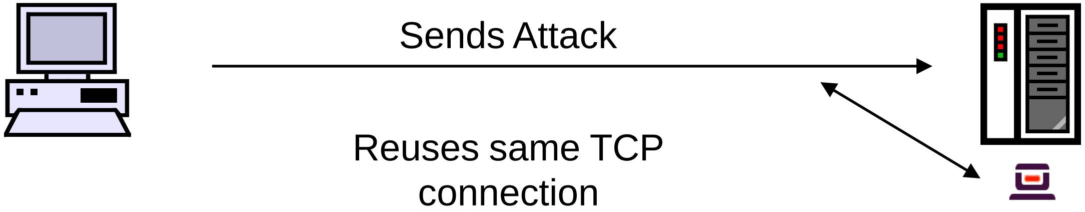
- Connect from target



Sends Attack

Connects FROM agent

- Reuse socket



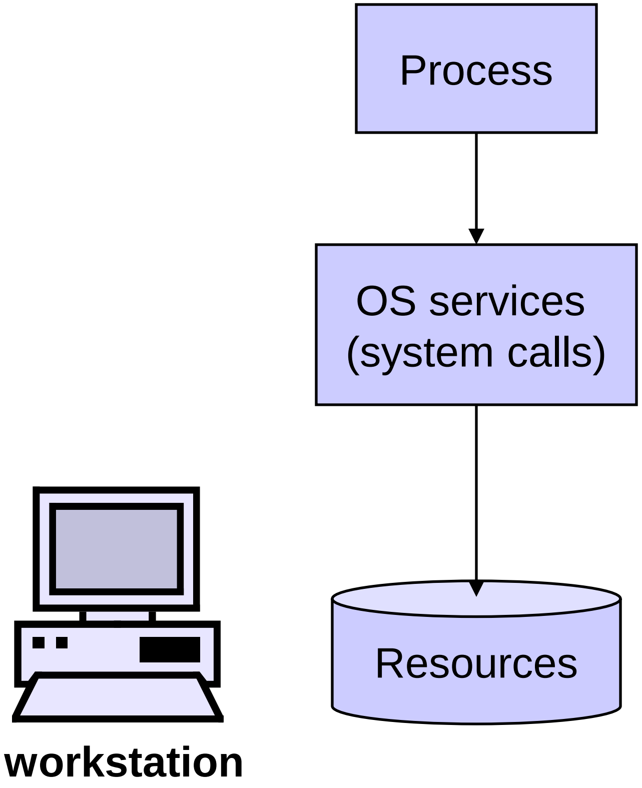
Sends Attack

Reuses same TCP connection

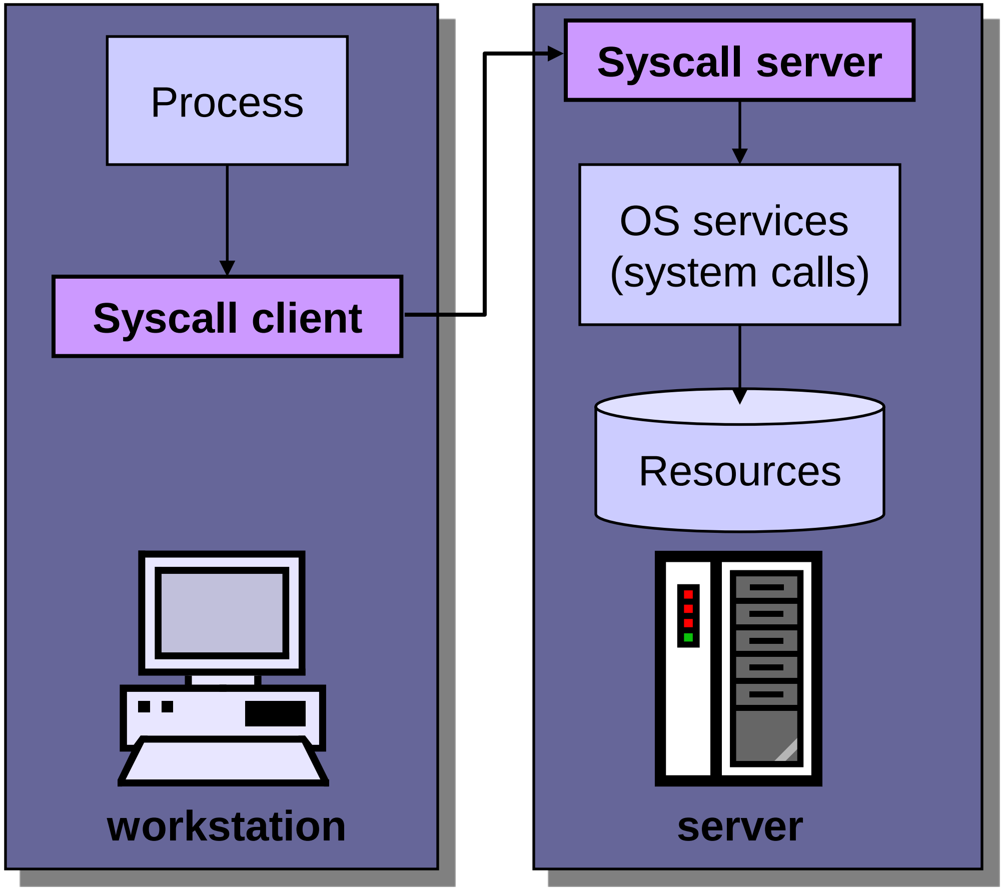
Syscall proxying agents transparently provide remote execution

SYSCALL PROXYING AT A GLANCE

- A process interacts with resources through the OS



- SysCall Proxying in action



References implementation of Syscall Proxying and Inline Egg areavailable

- Syscall Proxying

Windows and Linux x86 reference implementation for non-commercial use

http://www.coresecurity.com/files/files/13/SyscallProxying.pdf

http://www.coresecurity.com/files/files/13/Samples.zip

- Inline Egg

Reference implementation using Python for non-commercial use

http://community.corest.com/~gera/ProgrammingPearls/InlineEgg.html

OTHER ATTACK TARGETS

The list of possible targets of attacks is not limited to just servers and networking equipment

ATTACK TARGETS

- Routers, switches, servers, FWs, IDSes
- The organization as a whole
- Individuals and their workstations
- Other networking capable gadgets
- Trusted third parties
- … and more?

The whole organization as target

ORGANIZATION AS TARGET

- Publicly available information
- Business oriented targets
- Security beyond the perimeter
- An organization is dependant on people
- Physical security
- Denial of service – Public image attacks

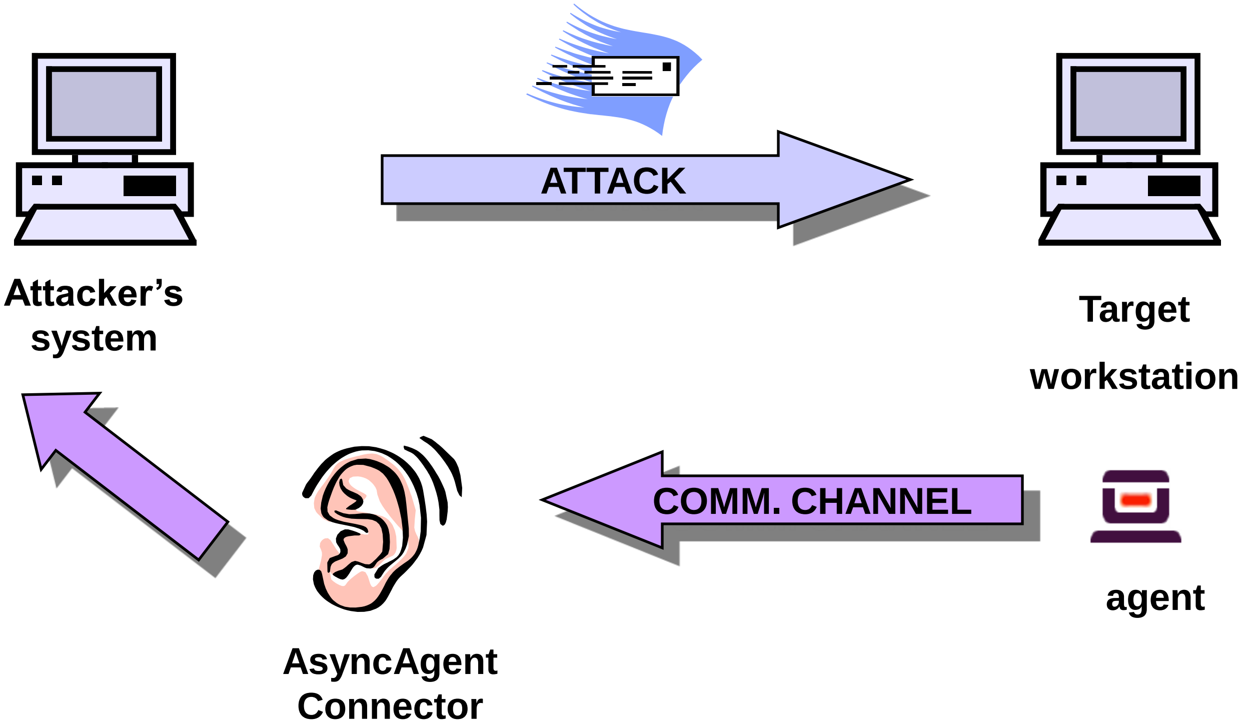
Attacks against specific individuals and their environment

PERSON AS TARGET

- Some examples
- Representations of a Person
- Impersonation attacks
- Use the front door (not the backdoor)
- Person - Workstation - Client side attacks
- Internal honeypots and IDSes

Attacking workstation software requires solution to some technical questions and implementation of a suitable framework

- Anatomy of a real–world client side attack.



Targeting individuals has several advantages

ADVANTAGES

- Lighter maintenance
- Less skilled enemy
- More software (more bugs)
- More targets
- Right to the inside
- Diversity is better

…but requires more sophisticated techniques and a flexible framework

DISADVANTAGES

- Tougher tuning
- It may be more noisy
- Asynchronous nature
- Communication channel
- Uptime

To effectively use persons as attack targets we need a whole new set of tools

TOOLS

- Network mapping using email headers
- Person discovery tools
- Craft profiles / trust relationships graphs
- OS and application detection
- Reverse traceroute

Network capable gadgets are also part of the infrastructure and therefore possible targets

NETWORK CAPABLE GADGETS

- Network printers
- Home DSL routers and cable modems
- Cellular phones, PDAs
- Gaming consoles, cameras
- Other embedded systems

More attack sophistication and efficiency can be gained by improving methodologies and applying problem-solving technologies

ATTACK PLANNING AND ATTACK MODELING

STARTING THE ATTACK

- Establish candidate target hosts
- Determine host liveness
- Network mapping
- OS Detection
- Identification of target services

How useful is the current methodology?

SOME QUICK QUESTIONS

- How do we use the outcome of IG?
- Do we use all the information we gather?
- Does it really matter if port 9 is open?
- Does it help to know the OS of every host?
- Is it really worth using a Vuln.Scanner?

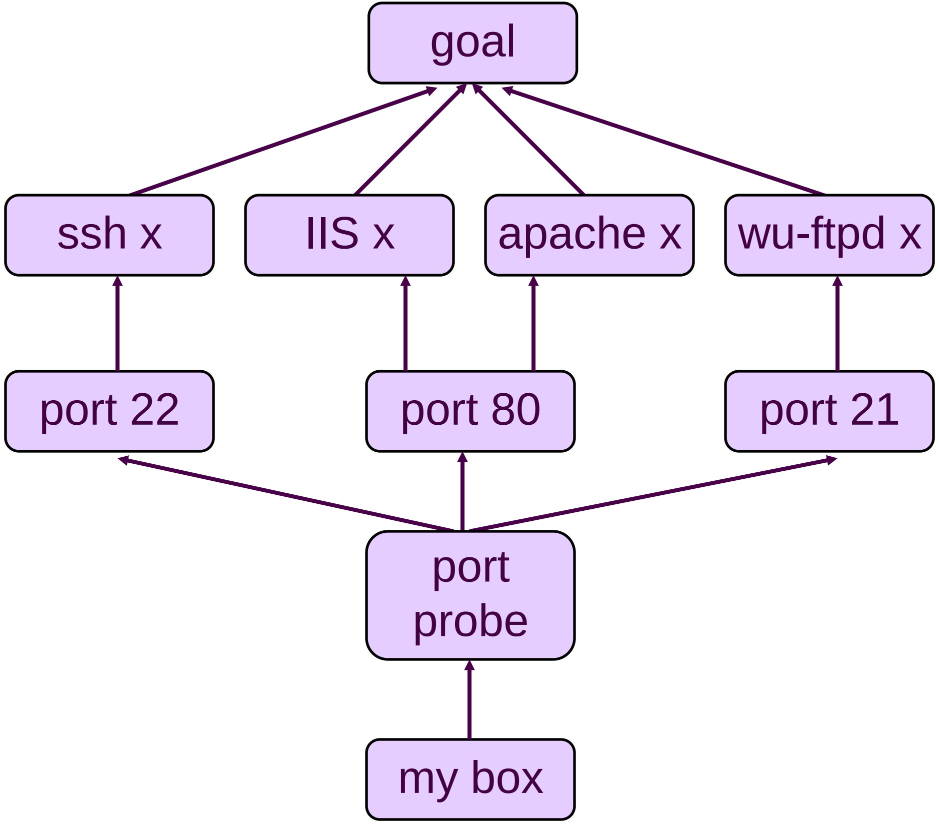
An example of attack planning for the information gathering phase

Goal: To gain control of any host in target network

Assets: Target's IP address
Control of my box
A set of IG tools and exploits

Actions:
test if a given port is open (port probe)
exploit ssh (on an OpenBSD)
exploit wu-ftpd (on a Linux)
exploit IIS (on a Windows)
exploit apache (on a Linux)

Plan:
Probe only ports 22, 80 and 21.
Probe port 80 first!
As soon as a port is found open, run an exploit.
Keep probing other ports only if exploit fails.



goal

ssh x

IIS x

apache x

wu-ftpd x

port 22

port 80

port 21

port probe

my box

Our simplistic example can outline some interesting lessons

INTERESTING NOTES

- Planning for tools we already have
- Planning for services on standard ports
- Simple goal
- Different priorities would influence the plan
- Do we really need to **port** probe?
- How could we use an OS detector?

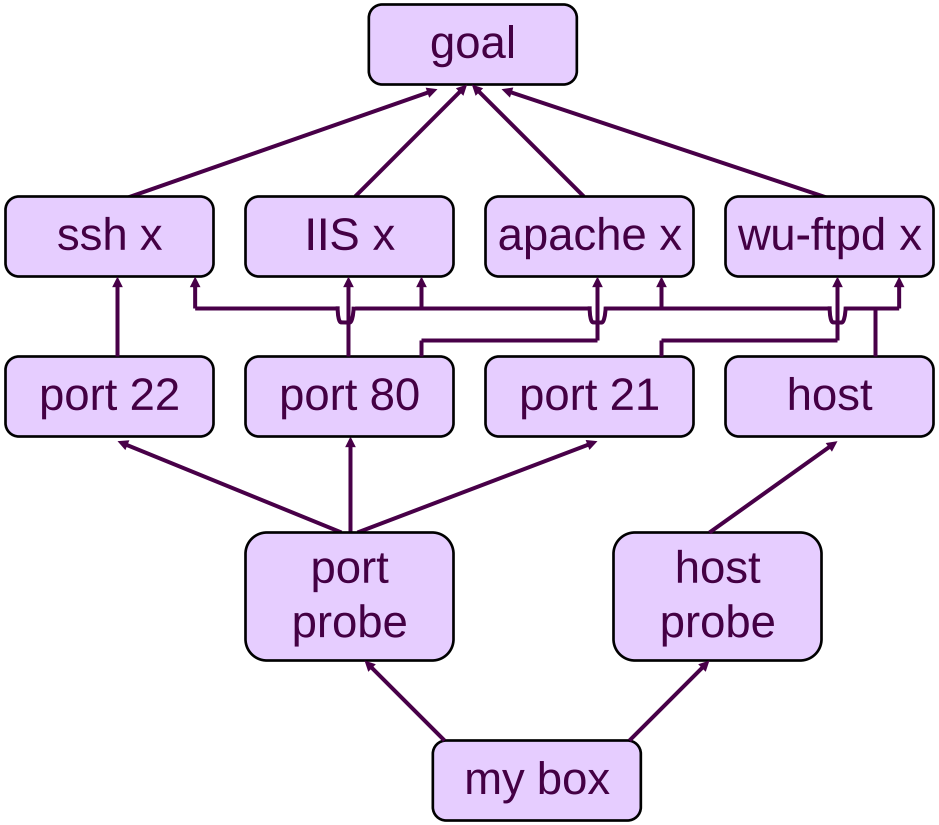
A slight variation of our first example…

Goal: To gain control of ALL possible hosts on a given network

Assets: Target's IP address
Control of my box
A set of IG tools and exploits

Actions:
test if a given port is open (port probe)
test if a given host is alive (host probe)
exploit SSH (on an OpenBSD)
exploit wu-ftpd (on a Linux)
exploit IIS (on a Windows)
exploit apache (on a Linux)

Plan:
Don’t use the host probe first.
Probe only ports 80,22 and 21
Probe ONLY port 80 first!
Launch exploit for every open port.
Probing other ports if exploit fails.
[Host probe remaining hosts]
[Probe non-standard ports]



… illustrates some common sense ideas

OTHER INTERESTING NOTES

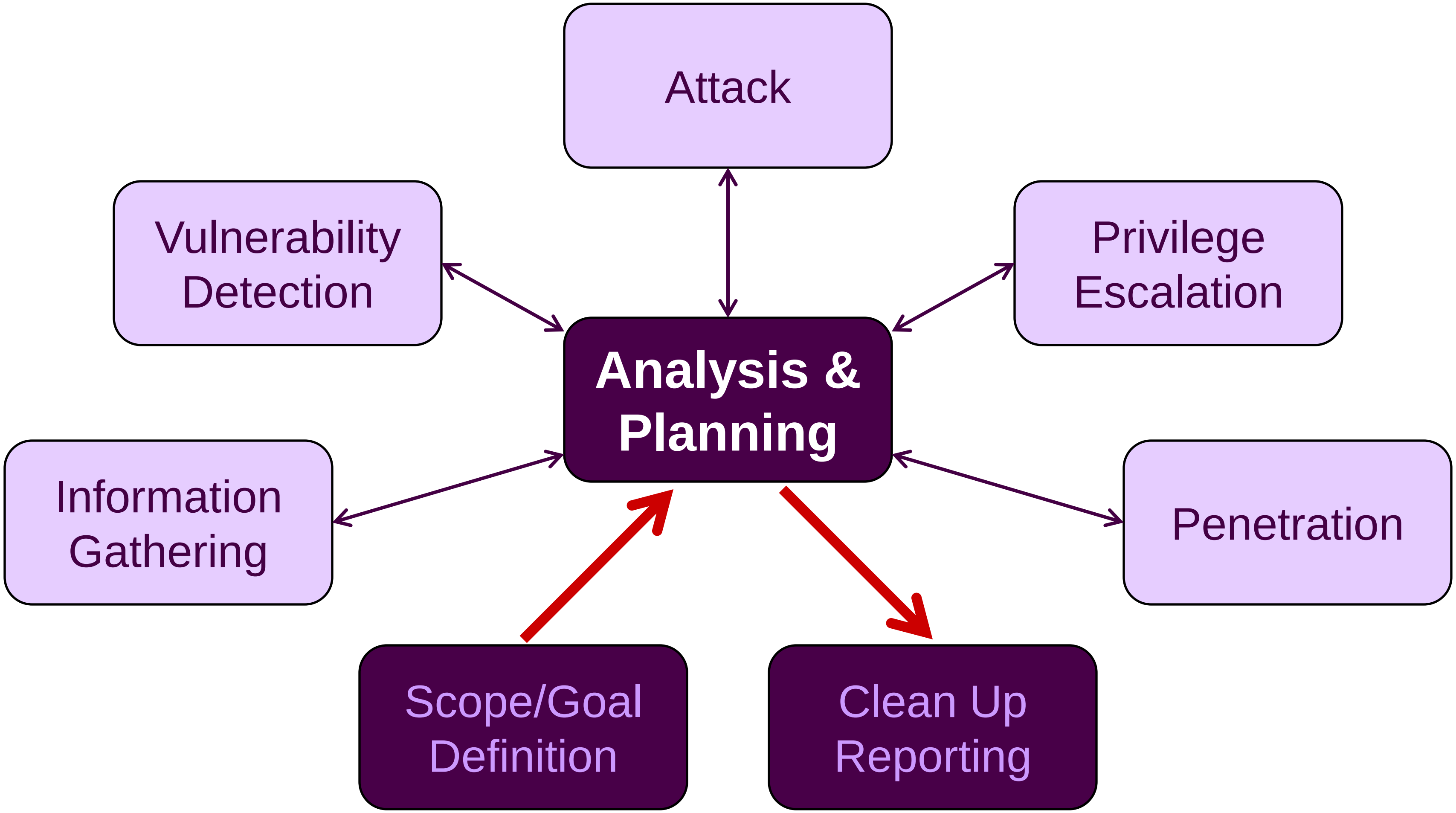
- The plan depends of the end goal
- Planning based on available assets
- Planning based on available information
- Kelyacoubian statistics, known ports
- Do we really need to **host** probe?
- How could we use an OS detector?

As the number of available tools increases the complexity of planning and executing successful attacks also increases

WHAT IF…

- Our IG and exploit tools are un-realiable
- Our exploit tools can disrupt targets (DoS)
- Some exploits have dependencies on others
- Goals are defined more precisely
- Systems and individuals detect attacks and react
- Attack execution time is constrained

Introduction of technology-based attack analysis and planning can solve some problems



To address attack analysis and planning we must first be able to model attacks from the attacks perspective

USES FOR AN ATTACK MODEL

- Attack planning
- Risk assessment
- Attacker profiling
- Higher level of abstraction for IDS
- Computer aided intrusion
- Automated intrusion
- Priorization of tool development

The model - *Introduction*

MODEL COMPONENTS

- Actions
  - Things you can do
- Assets
  - Things you have or know
- Agents
  - The actors, who can do **Actions**
- Goals
  - Purpose and end result of attack
- Costs
  - The cost of a given action
- Plan
  - Actions needed to fulfil a goal
- Attack Graph
  - Union of all possible plans

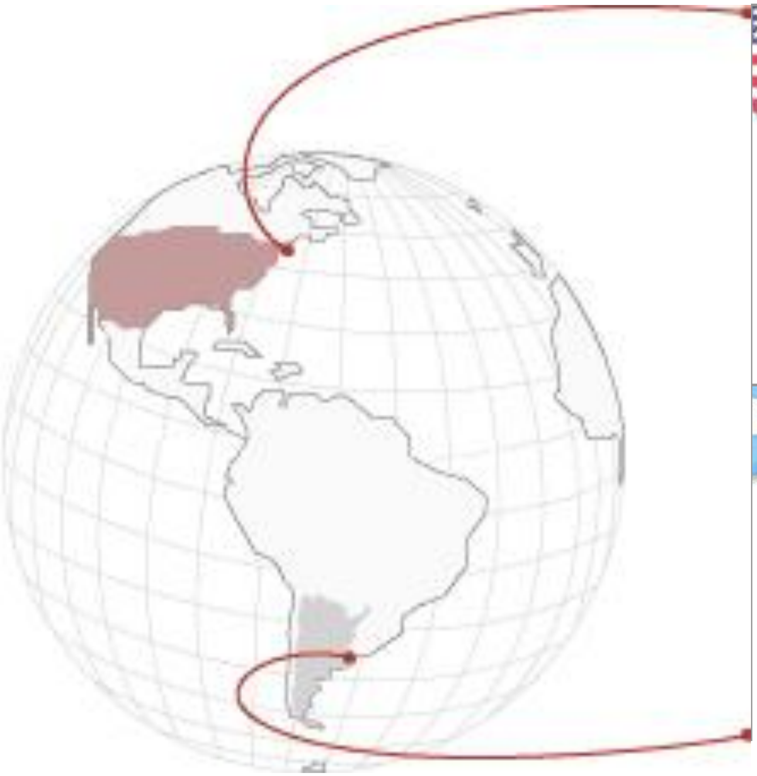
Existing models do not reflect the attacker’s concerns

SOME CONCERNS ASSOCIATED TO “COST OF ATTACK”

- Produced noise / Stealthiness
- Total running time
- Probability of success
- Trust
- Traceability
- Novelty (0-day-ness)

ivan.arce@coresecurity.com | gerardo.richarte@coresecurity.com

CONTACT INFORMATION FOR FOLLOW-UP



Headquarters · Boston, MA
46 Farnsworth St
Boston, MA 02210 | USA
Ph: (617) 399-6980 | Fax: (617) 399-6987
info@coresecurity.com

Research and Development Center
Argentina (Latin America)
Florida 141 | 2º cuerpo | 7º piso
(C1005AAC) Buenos Aires | Argentina
Tel/Fax: (54 11) 5032-CORE (2673)
info.argentina@coresecurity.com

www.coresecurity.com