



DSci526: Secure Systems Administration

**Configuration Management
Group Projects**

Prof. Clifford Neuman

Lecture 5
17 February 2021
Online



Course Identification

- DSci 526
 - Secure Systems Administration (4 units)
- Class meeting schedule
 - Usually 2PM to 5:20PM Wednesday
 - Online
- Class communication
 - dsci526@csclass.info
 - Goes to instructor and any assistants and is archived.



Announcement

- Alternate Schedule for Today
 - 10:30AM to 1:50PM
 - Same Zoom Link

Presentations: Configuration Management



- Marco Gomez
- Louis Uuh
 - We need to find a good date for this presentation, since the topics is covered in today's lecture.

February 24th – Red Teaming



- I encourage you to work together to prepare a joint presentation where each of you presents a different aspect of the topic.
 - Hanzhou Zhang
 - Yang Xue
 - Abhishek Tatti
 - Doug Platt
 - Shagun Bhatia
- Your group will have 100 minutes to present.
- This is Next Week. If your team requires more time to prepare, I can shift class topics around so that you can go a little bit later in the semester. But let me know (as a group) today.

March 3rd Presentations Secure Cloud Administration



- Secure Cloud Administration (20 min)
 - Sarahzin Chowdhury - Cloud Access Security Brokers
- Incident Response Planning (40 min)
 - Carol Varkey
 - Amarbir Singh

I encourage you to work together to prepare a joint presentation where each of you presents a different aspect of the topic.

March 17th – Secure Networking



-
- Christopher Samayoa (Network Access Control)
 - Shanice Williams – Network Monitoring – WireShark
 - Pratyush Prakhar – Web Penetration Tools

I encourage you to work together to prepare a joint presentation where each of you presents a different aspect of the topic. This group will have 1 hour to present.

March 31st – Security Incident Event Management



- Malavika Prabhakar
- Anthony Cassar
- Dwayne Robinson (Network Perimeter Detection)
- MaryLiza Walker (Attack Forensics)
- Jason Ghetian
 - I encourage you to work together to prepare a joint presentation where each of you presents a different aspect of the topic.
 - I will give your team 1:40 to present.



Linux Related Topics

- Azzam Alsaeed – SELinux
- Alejandro Najera – Linux Administration
- Tejas Pandey – Identity Management in Linux
- Ayush Ambastha – Linux Kernel Security
 - We need to select a week for this topic since we have already covered it in lecture.
 - I will give your team 1:20 to present.
 - I encourage you to work together to prepare a joint presentation where each of you presents a different aspect of the topic.



DSci526: Secure Systems Administration

Configuration Management
Group Projects

Prof. Clifford Neuman

Lecture 5
17 February 2021
Online

Configuration Management



Configuration management (CM) refers to a discipline for evaluating, coordinating, approving or disapproving, and implementing changes in artifacts that are used to construct and maintain software systems. An artifact may be a piece of hardware or software or documentation.

-- A Framework for Software Product Line Practice, Version 5.0, CMU Software Engineering Institute

Configuration Management (CM)



- Alternate Definition: CM refers to the process of systematically handling changes to a system in a way that it maintains integrity over time.
- Configuration Management can mean:
 - Network device configuration
 - Version of applications that are running
 - What patches have been installed on your system.
 - Hardware and software / application baselines for your devices/systems

Basic Configuration Management



- *Configuration Management (CM)* comprises a collection of activities focused on establishing and maintaining the integrity of products and systems, through control of the processes for initializing, changing, and monitoring the configurations of those products and systems.
- A *Configuration Item (CI)* is an identifiable part of a system (e.g., hardware, software, firmware, documentation, or a combination thereof) that is a discrete target of configuration control processes.
- A *Baseline Configuration* is a set of specifications for a system, or CI within a system, that has been formally reviewed and agreed on at a given point in time, and which can be changed only through change control procedures. The baseline configuration is used as a basis for future builds, releases, and/or changes.
- A *Configuration Management Plan (CM Plan)* is a comprehensive description of the roles, responsibilities, policies, and procedures that apply when managing the configuration of products and systems.

Benefits of configuration management

- Mass deployment and fast provisioning: CM tools automate efficient deployment/
- Roll back for recovery to a stable version in case of failure or compromise.
- Reduction of one-off server deployment, i.e. servers that are unintentionally different from an established configuration which may present security issues.
 - Minimization of alternative configurations.

Security and Configuration Management



- The configuration of an information system and its components has a direct impact on the security posture of the system.
 - Changes needed to keep up with threat landscape.
 - Changes affect previously established security posture
- Security-Focused Configuration Management (SCM) is the management and control of secure configurations for an information system to enable security and facilitate the management of risk.
- SCM builds on the general concepts, processes, and activities of configuration management by attention on the implementation and maintenance of the established security requirements of the organization and information systems.

Security and Configuration Management



- Identify and record configurations that impact the security posture of the information system and the organization.
- Consideration security risks in approving initial configuration.
- Analyze security implications of changes to information system configuration.
- Document and report approved/implemented changes.



Configuration Management

Know what is running

- [AFICK](#) or Tripwire
 - Manage history from another system

Keep your system up to date

- How depends on the size of your organization
- Standardize configurations
 - User machines, server machines
 - If you select patches, your systems should be configured to automatically follow your lead.

There is an industry around all of this

- Larger installations should use commercial products



Configuration Management

A process for consistently establishing and maintaining the characteristics of the components of a system relevant for the proper functioning of a system.

- Proper functioning includes:
 - Security
 - Updates and security patches.
 - Detection and prevention of unauthorized changes.
- Components includes all system assets:
 - Hardware
 - Software
 - Credentials
 - Licenses
- Characteristics includes:
 - Accounts
 - Settings
 - Policies.



Purpose of CM

- To Maintain Consistency of a system and its attributes with a technical baseline over the systems life.
- CM is part of system's security assurance cycle.
- Reduce the management workload for a collection of systems.
- Reduce the attack surface of a collection of systems by reducing the differences between individual systems within the collection.

What if Bad Configuration Management



Example Attack Scenarios for Misconfigured Systems

- An application server administration console was automatically installed and not removed.
 - An attacker discovered the standard admin pages and succeeded to log in with default passwords.
- Directory listing was left enabled on a server.
 - An attacker discovered the ability to list directories to find files on the server.



Any More Problems?

Out of Date?

Updates not applied

Vulnerable Apps?

Still running Flash.



It Starts with an Inventory

- Catalog of systems
 - What is approved for connection
 - Prevent access by uncatalogued systems
 - For each system:
 - Serial Number, Tag, MACs, IPs
 - Location, Owner, Admin
 - Make/Model, Hardware Features
 - Include routers, hubs, printers, other network attached items.
 - Purpose
 - Software (OS, patches, applications, etc)



It Starts with an Inventory

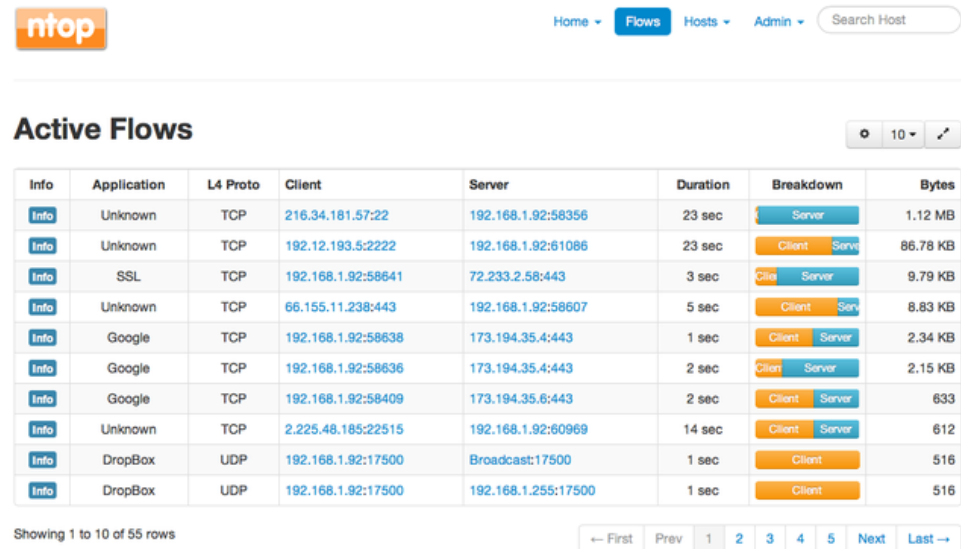
- Catalog of software
 - What is approved for use
 - Detect unauthorized installs
 - For each system:
 - Name, Version, Patch Level
 - Checksum
 - License information
 - System requirements
 - Security considerations/implications
 - Anything else



Local Network Management

NTOP – Network Monitoring

- Can be run on host or guest OS
- It is a major CPU hog but a useful tool
- Shows all network flows in real time
 - Useful to find flows used by applications
 - Which can then be restricted in iptables



Info	Application	L4 Proto	Client	Server	Duration	Breakdown	Bytes
Info	Unknown	TCP	216.34.181.57:22	192.168.1.92:58356	23 sec	Server	1.12 MB
Info	Unknown	TCP	192.12.193.5:2222	192.168.1.92:61086	23 sec	Client Server	86.78 KB
Info	SSL	TCP	192.168.1.92:58641	72.233.2.58:443	3 sec	Client Server	9.79 KB
Info	Unknown	TCP	66.155.11.238:443	192.168.1.92:58607	5 sec	Client Server	8.83 KB
Info	Google	TCP	192.168.1.92:58638	173.194.35.4:443	1 sec	Client Server	2.34 KB
Info	Google	TCP	192.168.1.92:58636	173.194.35.4:443	2 sec	Client Server	2.15 KB
Info	Google	TCP	192.168.1.92:58409	173.194.35.6:443	2 sec	Client Server	633
Info	Unknown	TCP	2.225.48.185:22515	192.168.1.92:60969	14 sec	Client Server	612
Info	DropBox	UDP	192.168.1.92:17500	Broadcast:17500	1 sec	Client	516
Info	DropBox	UDP	192.168.1.92:17500	192.168.1.255:17500	1 sec	Client	516

Showing 1 to 10 of 55 rows

← First Prev 1 2 3 4 5 Next Last →



Aspects of Configuration Management

- Organizational and Process
 - Change Management
 - Admission Policy
- Technical
 - Dependency Management
 - Patch Management (also organizational)
- Live evaluation and detection
 - Admission Control
 - Allowed Lists
 - Change detection



Organization aspects of CM

- Identify configuration items to be controlled
 - document user requirements, system design and development, software version, interface control, data flows and network diagrams, test plans and procedures.
 - Use schema or comply with organizational policy to provide unique identifiers for each item.
- Identify the level of CM to be controlled, and determine the hierarchy of each level
 - nature of the system, configuration items, components.
- Identify all baselines to be managed
 - user requirements, system requirements, design, development, experimentation, sustainment, etc.



Configuration Control

- Develop a process to track all configuration item changes and intrusion in appropriate baseline.
- Build or provide specifications to build work products from the software configuration management system, or physical products from the hardware configuration management system.
- Purchase or develop tools for version control of source code, providing version control tracking to the line of code level. Implement an engineering release system to provide hardware version control.



Change Management

- It is the documented process for managing and controlling changes to the configuration of an information system or its constituent CIs.
- It involves:
 - systematic proposal
 - Justification
 - Implementation
 - test/evaluation
 - Review
 - disposition of changes to the system, including upgrades and modifications.
- Configuration change control is applied to include changes to components of the information system, changes to the configuration settings for information technology products, emergency/unscheduled changes, and changes to remediate flaws.
- Changes are controlled from the time the change is proposed to the implementation and testing of the change
- The emphasis here is put on the management of change to maintain the secure, approved baseline of the information system



Configuration Status Accounting

- Publish periodic reports describing the current configuration of each item during the entire life of the system.
 - Hardware – include details
 - Software – include version
 - Other controlled assets
- Note level of “acceptance” testing performed.



Configuration Audits

- Performing periodic verifications of operational baselines for completeness.
- Assure that both functional and physical configuration meets the requirements.
- Tools, scripts or logs can be used when a change has occurred in a configuration.



Technical Aspects of CM

- Dependency Managers
 - Linux package managers
- Patch Management
 - Software update options
 - Software update center (linux)
 - Windows updates
 - App Stores
- Special Tools
 - Secuinia, others (later)
- New attack vectors
 - When to update



Tools: CM, evaluation and detection

- Live evaluation and detection
 - Admission Control
 - Allowed Lists
 - Change detection



Automated Tools: Ansible

- Free software developed by Red Hat
 - Written in Python, can run on multiple platforms
 - Playbook written in YAML to perform tasks
 - Managed servers do not require agents.



Automated Tools: Ansible

- Example handlers
 - Install WordPress configuration file, and restart httpd

tasks:

- name: install WordPress configuration file
template: src=wordpress.conf dest=/etc/httpd/conf.d/wordpress.conf

notify:

- restart httpd

handlers:

- name: restart httpd
service: name=httpd state=restarted

- Does this seem familiar?



Some other Tools

- Chef
 - Written in Ruby, can be easily customized.
 - Can run on multiple platforms.
- Puppet
 - Written in Ruby.
 - Can run on multiple platforms.
 - for large scale systems
- CFEngine
 - Can run on multiple platforms.
 - Describe the final state in which one wants to end up. The agent then ensures that the necessary steps are taken to end up.
- Microsoft PowerShell DSC
 - Standard since PowerShell 5.0 (Windows 8.1), desired state configuration



File Integrity Monitoring

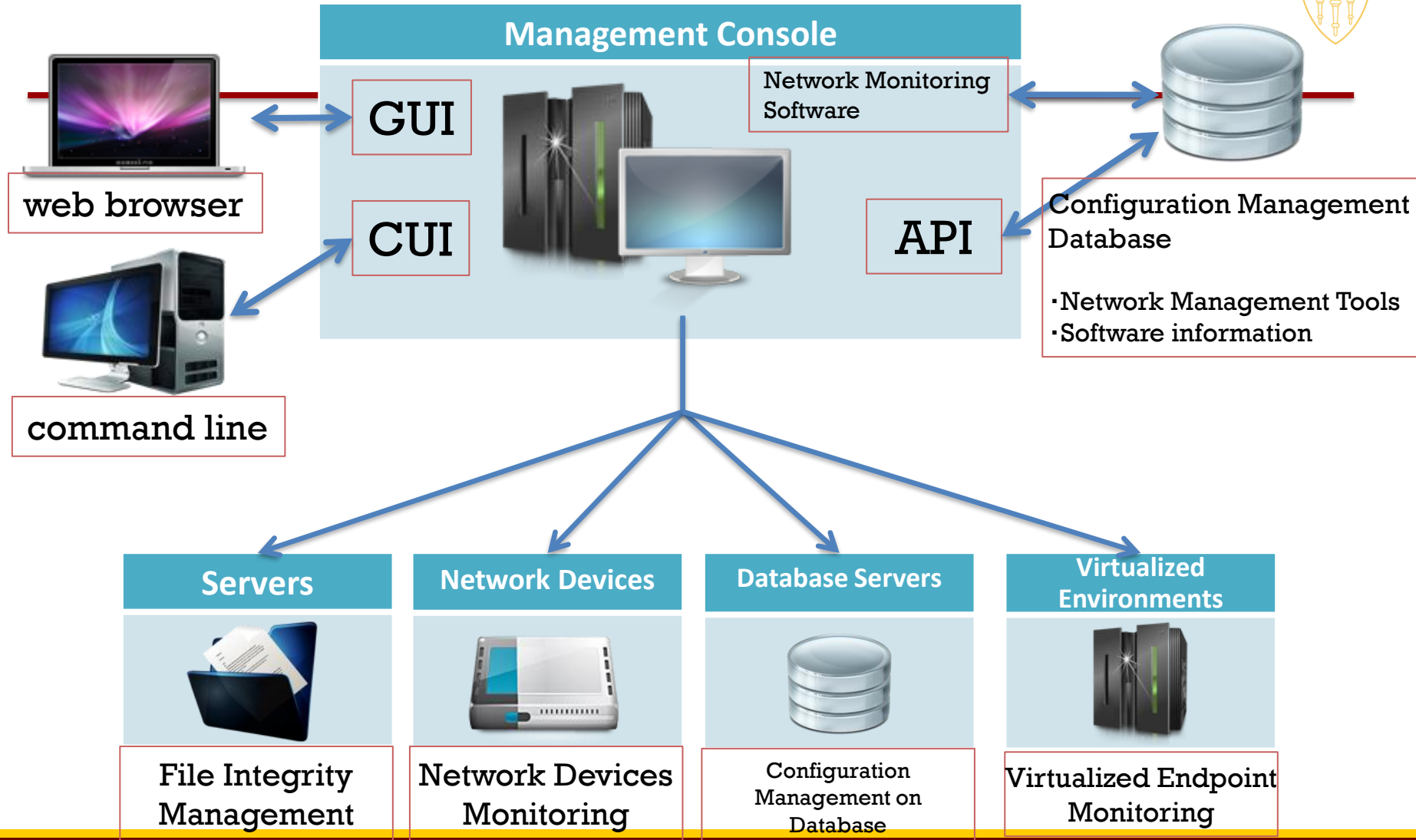
- Validate integrity of operating system and application software files
 - Generate Baseline – File Signatures (hashes)
 - Regenerate and compare to baseline.
 - Repeat periodically.



Automated Tools Detection: Tripwire

- Open source and enterprise developed by Tripwire, Inc.
 - Two versions, Open Source Version is not maintained or upgraded
 - Detect changes to file system objects
 - When first initialized, scans the file system
 - Stores information in a database.
 - To Verify
 - Same files are scanned
 - Results compared against the stored values in the database.
 - Changes are reported to administrator.
- Alternative for supported open source – AFIK
 - Another File Integrity CheckKer
 - Perl based, deployment on Windows, Linux, Unix, Solaris.
- AIDE
 - Advanced Intrusion Detection Environment
 - runs on Linux

Tripwire Enterprise





File Signature Bypass Issues

- Default MD5 signature can be bypassed by tools.
- Having a wide variety of simultaneous cryptographic generation algorithms can help to detect evasion through signature weaknesses.



Hash Algorithms for File Signatures

	md5	sha1	sha256	sha512	md160	tiger	whirlpool	gost	crc32	haval
Tripwire	✓	✓	✓	✓						
AFICK	✓	✓							✓	✓
AIDE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓



Protecting the Tools

- **Tripwire**
 - Require two passphrases longer than eight characters
- **AFICK**
 - Calculate MD5 signature of itself right after the first installation
 - Can boot from CDROM
- **AIDE**
 - Signed and stored in the Ubuntu repository, automatically verified during the download and installation



Placement of CM Functions

- Central to the Enterprise
- On each system
- In the Network
- On storage or other servers
- Relationship to SIEM



Minimization in CM

- Create a few standard configurations
 - Fewer different systems to configure and possibly get wrong.
- Automate the configuration of machines within groups.
 - So that you don't leave one out of the update cycle



Push vs Pull

- Most end user systems pull updates
 - Windows updates
 - Linux software updater
- Much server infrastructure changes based on a push model.
 - Ansible (red hat)

Tools: Secunia CSI (CSI Corporate Software Inspector)

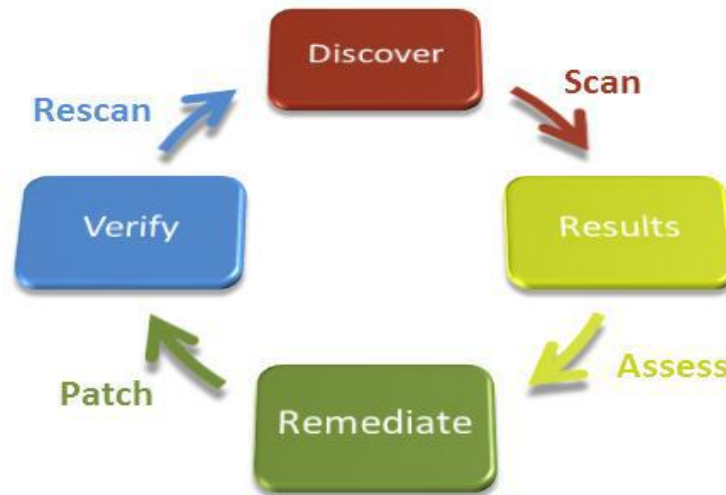


- Windows based Vulnerability and Patch Management Software Solution that completes and targets the Patch Management process.
- Combines Vulnerability Intelligence, Vulnerability Scanning, and Patch Creation with Patch Deployment Tool Integration
- Collects metadata from primarily .EXE, .DLL, and .OCX files on the system being scanned and sendst to Secunia's Secure Data Processing Cloud where it is processed and parsed.
- Compared against Secunia File Signatures, metadata to actual product installation resulting in exact version extracted from metadata
- Inventory is then compared against the unique Secunia Advisory and Vulnerability Database

Tools: Secunia CSI (CSI Corporate Software Inspector)



- The result is a precise inventory of products, their versions, the security state of each, and a detailing of vulnerabilities and their criticality and impact.





Tools: SaltStack

- Python-based open-source configuration management software and remote execution engine for Linux and Windows platforms.
- Configuration management system called Salt States
- Salt functions on a lead/follower topology. A lead server acts as a central control bus for the clients, which called minions. The minions connect back to the leader.
- Also runs in a standalone mode
- Provides configuration management functions by automating the packaging and provisioning of code into an organization's operational IT environment.
- Can use scripts written directly in Python, or can render other scripts, such as those written in YAML or JSON, through the use of the PyDSL Salt renderer
- Capable of storing configuration directives, and then instructing other machines to follow those directives by installing software, making configuration changes to the software, and reporting back on the progress and succes or failures.

SaltStack CM scenario: installing LAMP stack on Red Hat



- SaltStack formulas and states can be used for tasks such as installing a package, configuring and starting a service, setting up users or permissions, and many other common tasks.
- E.g. set up Apache web server and MySQL database server for the web application.

- define the database server first

```
# /srv/salt/mysql.sls
mysql:
  pkg.installed:
    - name: mysql-server
  service.running:
    - enable: True
    - require:
      - pkg: mysql-server
```


SaltStack CM scenario: installing LAMP stack on Red Hat



- Apache installation is more complex, because it includes a configuration file. It is copied to web server using file.managed function , which supports enhanced functionality such as templating. To accommodate this, create an apache/ directory inside of /srv/salt/

```
# /srv/salt/apache/init.sls
```

```
httpd:
```

```
  pkg.installed:
```

- name: httpd
- file: httpd

```
  service.running:
```

- enable: True
- require:
- pkg: httpd

```
  file.managed:
```

- name: /etc/conf/httpd/httpd.conf
- source: salt://httpd/httpd.conf
- require:
- pkg: httpd

SaltStack CM scenario: installing LAMP stack on Red Hat



- ```
/srv/salt/top.sls
base:
 web*:
 - apache
 db*:
 - mysql
```
- This definition will ensure that any servers whose names start with “web” (such as web01 or even web01.example.com) will have the Apache state applied to them
- and any servers whose names start with “db” (such as db01 or db01.example.com) will have the MySQL state applied to them.
- To apply these states to all servers:  

```
salt '*' state.highstate
```
- A highstate is the combination of state data (packages, services, files, etc.) that will be applied to the target system

# SaltStack: Addressing configuration drift

---



- If `httpd.conf` file is changed on server, SaltStack resets it and reports to administrator what changes were made to enforce the correct state.
- But what about package versions?
  - When `pkg.installed` state is declared, SaltStack checks with underlying package manager to see if package is already installed.
    - If so, state has been archived and no action needed.
    - If Not, it has package manager install most recent version of package
  - Allows one to lock down version to specific version



# DSci526: Secure Systems Administration

First Group Project

*Prof. Clifford Neuman*

**Lecture 5**  
17 February 2021  
Online



# Teams for First Group Project

---

- Team One

- Shagun Bhatia
- Anthony Cassar
- Sarahzin Chowdhury
- Aditya Goindi
- Tejas Kumar Pandey
- Malavika Prabhakar
- Pratyush Prakhar
- Dwayne Robinson
- Christopher Samayoa
- Amarbir Singh
- Louis Uuh
- Shanice Williams

- Team Two

- Azzam Alsaeed
- Ayush Ambastha
- Jason Ghetian
- Marco Gomez
- Alejandro Najera
- Doug Platt
- Abhishek Tatti
- Carol Varkey
- MaryLiza Walker
- Yang Xue
- Hanzhou Zhang



# Banking Scenario

- Your organization must:
  - Maintain a database of account holders
  - A database of account balances
  - Enable web access by customers who:
    - Can update their personal information
    - Check their account balance
    - Transfer funds to another account (by number)
    - View transactions on their account
    - Submit an image of a check for deposit
      - (check should be viewable, but you do not need to scan it or process it)
- Access is needed
  - Via web from the open internet
  - Outbound email confirming transactions
  - All other interactions may be limited by information flow policies to internal machines.



# Teams

---

- You already enumerated data and users and suggested a containment architecture.
- There are 11 or 12 members per team
- Teams have 4 weeks to complete project
- Teams should decide on sub-groups
  - Based on skill sets
  - 2 or 3 members per sub-group
  - These members will focus on different aspects of deployment.



# In your Breakout Groups

- Decide on a team name
- Decide on Tasks for sub-groups, I'd suggest:
  - Platform administration (including configuration management)
  - Network administration and deployment
  - Firewall / VPN administration and deployment
  - Server Development and deployment (web server)
  - Server Development and deployment (back-end, database)
  - Intrusion Detection/SIEM
  - Red Teaming and Penetration Testing
- Share (by email) last weeks assignment
- Discuss (by email) combined architecture
- A spokesperson (different each week) will report for 10 minutes to the entire class on your progress each week (starting next week)
  - All can chip into the discussion, but the spokesperson will lead it.
  - Teams will provide a progress update, but will likely withhold some information if it provides an advantage.
- Today try to organize into roles in terms of who will do what for development and deployment.
  - With 11 or 12 members, there should be 2 or 3 team members in each role (i.e. you will have sub-teams)



# Group Exercise One



- Decide on the software components to be deployed to implement software requirements on next slide.
  - Custom development should be simple scripts.
  - Use packages for database and other components.
- Decide on the VM's to be created to run those software components.
  - You can run more than one software component within a VM if you choose.
  - Decide on the methods you will use to contain access to those software components, and to the information managed by those components.
- Configure communication between VM's and to the outside
- Install packages
- Write scripts and demonstrate basic flow through system.
- Report on progress as group before class on February 24<sup>th</sup>.



# **INF526: Secure Systems Administration**

**Penetration Tools  
(advance slides for 2/24/21 lecture)**

***Yatin Wadhawan (Ph.D. candidate)***

***Prof. Clifford Neuman***



# DISCLAIMER

---

DO NOT USE THESE TOOLS AND METHODOLOGY FREELY OVER THE INTERNET. IT MAY CAUSE DAMAGE TO SOME ORGANIZATION'S CYBER INFRASTRUCTURE WHICH IS A CRIMINAL OFFENCE. THIS TUTORIAL IS JUST FOR LEARNING PURPOSE.

**AUTHORS DO NOT ENCOURAGE ANY MALICIOUS ACTIVITIES.**



# Topics

---

1. Ethical Hacking
2. Types of Hackers
3. Ethical Hacking Methodology
4. Information Gathering
5. Scanning
6. Attacks
7. Tools
8. Case Study: Ukraine Power Grid Attack



# Ethical Hacking

---

- **Primary motive:** To identify the weaknesses of the cyber infrastructure of an organization before an unethical hacker does.
- It is legal given testers have taken permission from the relevant stakeholders of the assets on which testing is performed.
- It is a subset of an Organization's security program.
- It does not just protect the information but helps organizations to **succeed**.



# Types of Hackers

---

- By Legality
  - Black Hat
  - White Hat
  - Grey Hat
- By Knowledge
  - Script Kiddies
  - Motivated Attackers
  - Coders
- By Motive
  - Criminals
  - Hacktivist
  - Governments

# Ethical Hacking Methodology



# Ethical Hacking Methodology







# Information Gathering

---

- Focused on collecting as much information as possible about the organization you want to compromise.
- Motive is to identify the entry and exit points.
- **Basic Methods:**
  - **Passive**
    - WHOIS, NSLookup etc.
    - Google Dork
    - DNS Info gathering
    - Social Engineering
  - **Active**
    - Ping
    - Traceroute

# Information Gathering (cont.)



- **Passive Methods**

- To gain information about targeted organization's cyber infrastructure without actively engaging with the systems.

- **WHOIS**

- Anyone can use the this service to search for databases and identify the registrant of a domain name and other information.
- It also provides the information regarding: IP address, name servers, admin contact etc.
- Link: <http://whois.domaintools.com/>

# Information Gathering (cont.)



- **Google Dork**

- It uses Google search engine to find security holes on the web applications over the internet.
- To locate specific strings of text within search results.
- Link: <https://www.exploit-db.com/google-hacking-database/>
- Some of the Operators
  - inurl .php?id=
  - intitle text
  - site text
  - filetype pdf

# Information Gathering (cont.)



- **DNS Information Gathering**

| Resource Records | Description                 |
|------------------|-----------------------------|
| A                | Return IPv4                 |
| AAAA             | Return IPv6                 |
| MX               | Mail Exchange Server        |
| NS               | Name servers                |
| AXFR             | Authoritative zone transfer |
| IXFR             | Iterative zone transfer     |
| SOA              | Start of the authority      |

# Information Gathering (cont.)



- **DNS Information Gathering**
  - ***dnsenum***: Tool in the backtrack Kali OS. It starts querying DNS servers and gather information:
    - Host address
    - Name servers
    - MX records
    - Gathering SOA records
    - Command: ***perl dnsnum.pl [host]***
  - ***dnsrecon***: to gather network infrastructure information.
  - ***Dig***: DNS information groper
    - ***dig example.com MX @ns0.example.com***

# Information Gathering (cont.)



- **Active Methods**

- Interact directly with a system of interest.

- **Ping**

- It is used to test the reachability of a system.
- It works at the network layer.
- It measures RTT, report errors and packet losses.
- One can also fix the size of the parameters using -l and number of request using -n.
- **Command:** ping -c 5 [www.example.com](http://www.example.com)
- **Result:** 64 bytes from xx.xxx.xxx.xxx: icmp\_seq=0 ttl=100 time=23.82 ms

# Information Gathering (cont.)



- **Traceroute**

- It is used to gather information about network infrastructure and IP ranges of a given host.
- Tool for displaying the overall path hop by hop from source to the destination.
- By default it sends the UDP packets.
- We can modify the command to send TCP/SYN and ICMP requests.
- **\$ traceroute -w 3 -q 1 -m 16 example.com**
- **\$ traceroute -I -w 3 -q 1 -m 16 example.com**
- **\$ traceroute -T -w 3 -q 1 -m 16 example.com**



# Information Gathering (cont.)

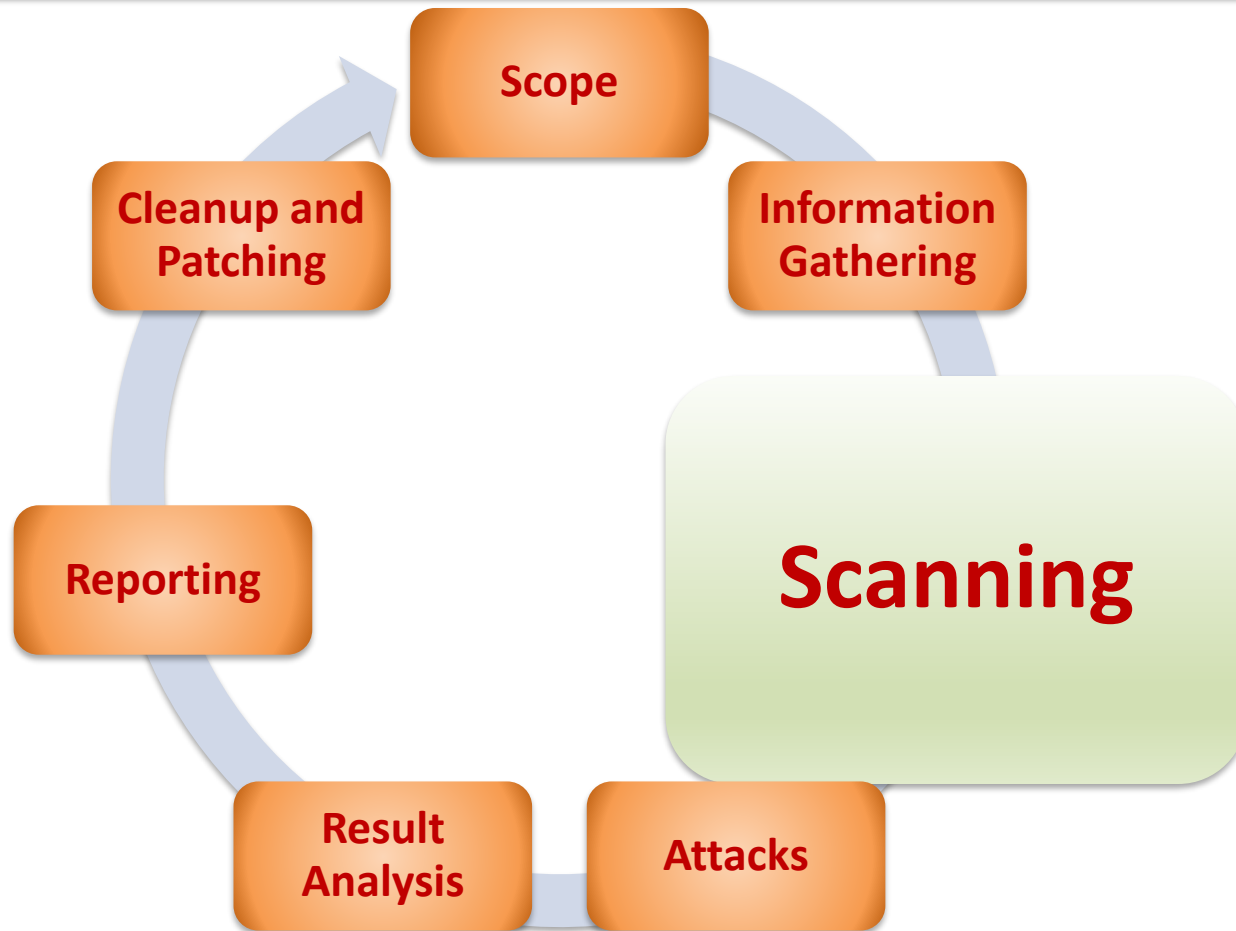
- Source: <http://www.inmotionhosting.com/support/website/how-to/read-traceroute>

```
C:\>tracert www.example.com
Tracing route to example.com [10.10.242.22]
over a maximum of 30 hops:

 1 <1 ms <1 ms <1 ms 172.16.10.2
 2 * * * Request timed out.
 3 2 ms 2 ms 2 ms vbchtmnas9k02-t0-4-0-1.coxfiber.net [216.54.0.29]
 4 12 ms 13 ms 3 ms 68.10.8.229
 5 7 ms 7 ms 7 ms chndbbr01-pos0202.rd.ph.cox.net [68.1.0.242]
 6 10 ms 8 ms 9 ms ip10-167-150-2.at.at.cox.net [70.167.150.2]
 7 10 ms 9 ms 10 ms 100ge7-1.core1.nyc4.he.net [184.105.223.166]
 8 72 ms 84 ms 74 ms 10gr10-3.core1.lax1.he.net [72.52.92.226]
 9 76 ms 76 ms 90 ms 10g1-3.core1.lax2.he.net [72.52.92.122]
10 81 ms 74 ms 74 ms 205.134.225.38
11 72 ms 71 ms 72 ms www.inmotionhosting.com [192.145.237.216]
```



# Ethical Hacking Methodology





# Scanning

- Till now we have understood how to create a profile of the target organization by finding the network information
- Now we need to find information about the specific IP addresses that can be accessed over the Internet, OS, accessible ports, network architecture, services running etc.
- Types of scanning:
  - Network
  - Port
  - Vulnerability



# Scanning (cont.)

---

- **Network Scanning**

- Tool to find out active host on the network
- You select the range of IP addresses and start scanning over the network.
- It provides the information Network devices including FTP servers and workstations.

- **Tools:**

- Advance IP scanner (Windows, Mac and Linux)
- Network Mapper (Nmap, ZenMap)
- Nessus



# Scanning (cont.)

Source: <http://angryip.org/screenshots/>

| IP            | Ping  | Hostname           | Ports [15+]  | Web detect               |
|---------------|-------|--------------------|--------------|--------------------------|
| 172.28.43.206 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.207 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.208 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.209 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.210 | 0 ms  | [n/a]              | 21,80        | CANON HTTP Server Ver2.2 |
| 172.28.43.211 | 0 ms  | [n/a]              | 21,80        | CANON HTTP Server Ver2.2 |
| 172.28.43.212 | 0 ms  | [n/a]              | 21,23,80,443 | [n/a]                    |
| 172.28.43.213 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.223 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.224 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.225 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |
| 172.28.43.226 | 0 ms  | pcee033219.int.han | [n/a]        | [n/a]                    |
| 172.28.43.227 | [n/a] | [n/s]              | [n/s]        | [n/s]                    |

Ready      Display: All      Threads: 0



# Scanning (cont.)

---

- **Port Scanning**

- Tool to find out which number of ports are accessible on a server or a host.
- Port scanning identifies open doors to a hosts.
- Nmap classifies port in these States:
  - Open
  - Closed
  - Filtered
  - Unfiltered



# Scanning (cont.)

---

- Commands:
  - nmap ipaddress
  - nmap -PN ipaddress
  - nmap -6 ipaddress
  - nmap -sP ipaddress (ping scan)
  - nmap --open ipaddress
  - nmap -p T:80 ipaddress
  - nmap -o ipaddress



# Scanning (cont.)

- Scan examples:
  - TCP SYN (-sS)
    - Half TCP connection
  - UDP Scan (-sU)
    - Send UDP packets
  - TCP NULL (-sN)
    - Does not set any bits (TCP flag header is 0)
  - TCP FIN (-sF)
    - Sets just the TCP FIN bit
  - TCP Xmas (-sX)
    - Sets the FIN, PSH, and URG flags
  - OS Detection (-o)



# Scanning (cont.)

- TCP NULL (-sN)

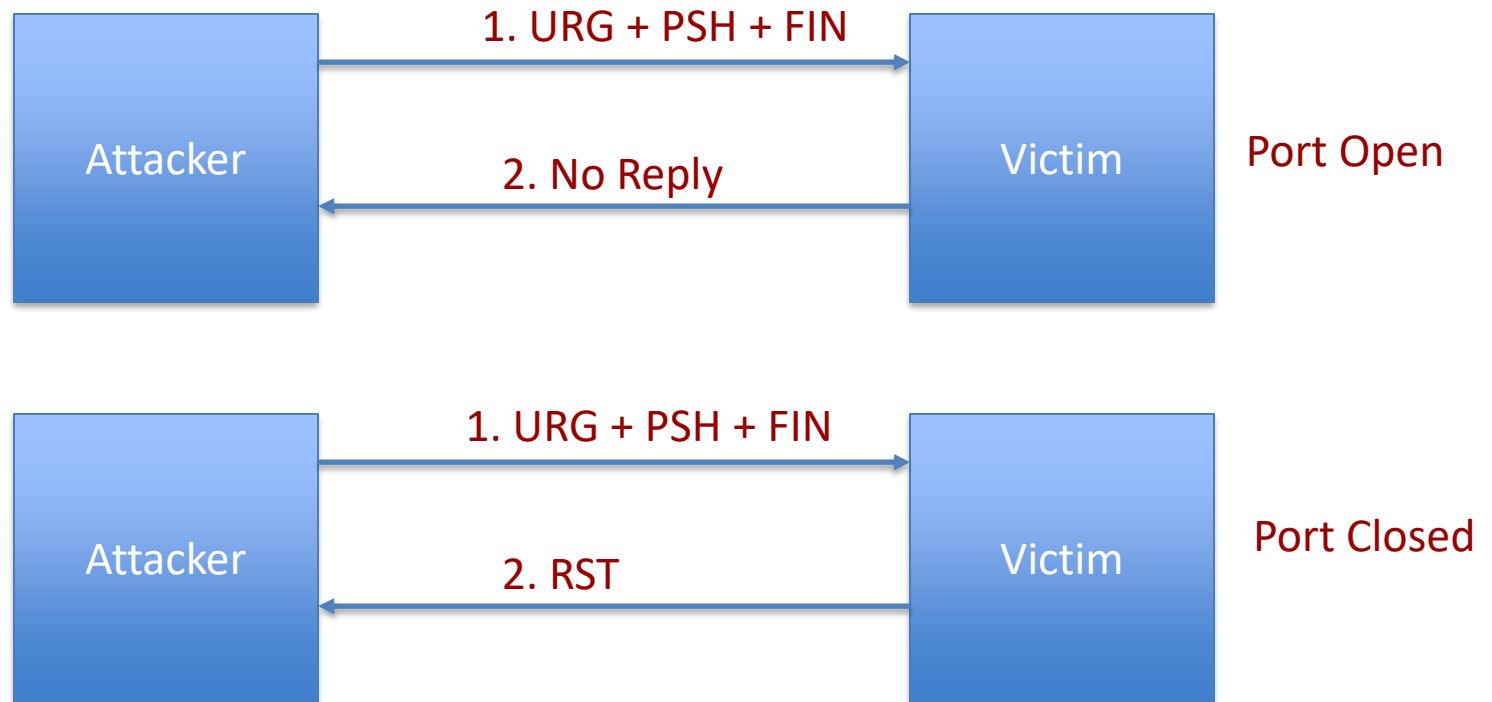






# Scanning (cont.)

- TCP XMAS (-sX)





# Scanning (cont.)

- -O Scan

```
nmap -O -v scanme.nmap.org

Starting Nmap (http://nmap.org)
Nmap scan report for scanme.nmap.org (74.207.244.221)
Not shown: 994 closed ports
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
646/tcp filtered ldap
1720/tcp filtered H.323/Q.931
9929/tcp open nping-echo
31337/tcp open Elite
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
OS details: Linux 2.6.39
Uptime guess: 1.674 days (since Fri Sep 9 12:03:04 2011)
Network Distance: 10 hops
TCP Sequence Prediction: Difficulty=205 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: /usr/local/bin/../../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.58 seconds
Raw packets sent: 1063 (47.432KB) | Rcvd: 1031 (41.664KB)
```

- Source: <https://nmap.org/book/osdetect-usage.html>



# Scanning (cont.)

- **Vulnerability Scanning**

- Once we have identified the accessible ports and services running on them, now we need to find the vulnerabilities associated with those applications.
- Tools:
  - Web Application **Acunetix, BurpSuite etc.**
  - Network Security **Nessus**
  - Mobile Security **Veracode, Tenable Security etc.**
- **Web Goat**
  - Insecure web application maintained by OWASP designed to teach web application security lessons.



# Scanning (cont.)

- **OWASP Top 10**

- Injection
- Broken Authentication and Session Management
- Cross Site Scripting (XSS)
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Missing Function Level Access Control
- Cross Site Request Forgery (CSRF)
- Using Components with Known Vulnerabilities
- Unvalidated Redirects and Forwards



# Scanning (cont.)

- Acunetix

## Add Scan Target

### General

Name

Description

IP / URL

OR

Choose a test domain ▼

Add Scan Target

testasp.vulnweb.com  
testaspnet.vulnweb.com  
testhtml5.vulnweb.com  
testmetasploitable.vulnweb.com  
testphp.vulnweb.com



# Scanning (cont.)

## Scan Targets

| <input type="checkbox"/> Name ^       | Host ⇅                                | Security | Status                       |                            |
|---------------------------------------|---------------------------------------|----------|------------------------------|----------------------------|
| <input type="checkbox"/> Test Scan 2  | http://testasp.vulnweb.com            |          | Web and Network Scans (Demo) | <a href="#">Scan Now</a> ▼ |
| <input type="checkbox"/> Test Scan 3  | http://testmetasploitable.vulnweb.com |          | Web and Network Scans (Demo) | <a href="#">Scan Now</a> ▼ |
| <input type="checkbox"/> Testing Scan | http://testphp.vulnweb.com            |          | Web and Network Scans (Demo) | <a href="#">Scan Now</a> ▼ |

1—3 of 3

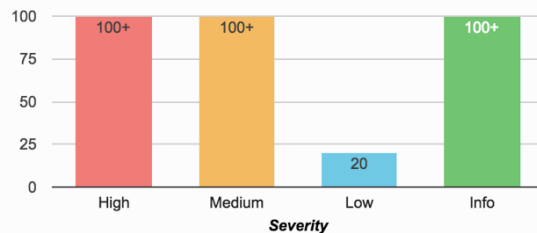


# Scanning (cont.)

## Dashboard

☐ Auto Refresh [Getting Started Wizard](#) [Documentation](#)

### Vulnerabilities by Severity



### Top 10 Vulnerabilities

|                                   |    |
|-----------------------------------|----|
| SQL injection (verified)          | 27 |
| Blind SQL Injection               | 27 |
| Cross site scripting (verified)   | 23 |
| Email address found               | 17 |
| Directory listing                 | 14 |
| Possible Trojan horse(s) detected | 9  |
| Broken links                      | 7  |
| Application error message         | 6  |
| Error message on page             | 5  |
| HTML form without CSRF protection | 5  |

### Latest Scans

| Host         | Type    | Threat | Completed    |
|--------------|---------|--------|--------------|
| Test Scan 3  | Web     | High   | 12 Jun 23:51 |
| Test Scan 3  | Network | High   | 12 Jun 23:51 |
| Test Scan 2  | Web     | High   | 12 Jun 23:47 |
| Test Scan 2  | Network | Medium | 12 Jun 23:47 |
| Testing Scan | Web     | High   | 12 Jun 23:46 |

### Most Vulnerable Hosts

|              |
|--------------|
| Testing Scan |
| Test Scan 3  |
| Test Scan 2  |

### Upcoming Scans

No upcoming scans




# Scanning (cont.)
















Alerts (2015) Knowledge Base (7)

537 139 65 1274 Generate Report

|                                     |                        |                                     |                                                                                                            |
|-------------------------------------|------------------------|-------------------------------------|------------------------------------------------------------------------------------------------------------|
| <b>Start Date</b> 12 Jun 2016 23:51 | <b>Files</b> 834       | <b>Requests</b> 833992              | <b>Host Name</b> <a href="http://testmetasploitable.vulnweb.com">http://testmetasploitable.vulnweb.com</a> |
| <b>End Date</b> 12 Jun 2016 23:51   | <b>Directories</b> 117 | <b>Avg. Response Time</b> 407.03 ms | <b>Scan Target Name</b> Test Scan 3                                                                        |
| <b>Duration</b> 0h 0m 5s            | <b>Variations</b> 701  | <b>Responsive</b> Yes               | <b>Scan Type</b> Web                                                                                       |

 AcuSensor was not detected during scanning.

 Demo scan results.

| Name                                                                                                                                           | Module                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| +  Code execution (17)                                        | Scripting (Code_Execution.script)                    |
| +  Cross site scripting (12)                                  | Scripting (XSS.script)                               |
| +  Cross site scripting (verified) (472)                      | Scripting (XSS.script)                               |
| +  Directory traversal (12)                                   | Scripting (Directory_Traversal.script)               |
| +  File inclusion (12)                                        | Scripting (File_Inclusion.script)                    |
| +  PHP-CGI remote code execution (2)                          | Scripting (PHP_CGI_RCE_Force_Redirect.script)        |
| +  Script source code disclosure (1)                          | Scripting (Script_Source_Code_Disclosure.script)     |
| +  Security vulnerability in MySQL/MariaDB sql/password.c (1) | Scripting (PHPInfo.script)                           |
| +  Server side request forgery (1)                            | Scripting (Server_Side_Request_Forgery.script)       |
| +  SQL injection (7)                                          | Scripting (Sql_Injection.script)                     |
| +  Apache 2.x version older than 2.2.9 (1)                  | Scripting (Version_Check.script)                     |
| +  Apache httpd remote denial of service (1)                | Scripting (Version_Check.script)                     |
| +  Apache httpOnly cookie disclosure (1)                    | Scripting (Apache_httpOnly_Cookie_Disclosure.script) |
| +  Application error message (18)                           | Scripting (Generic_Oracle_Padding.script)            |
| +  Cross site scripting (content-sniffing) (1)              | Scripting (XSS.script)                               |





# Scanning (cont.)

---

- CMS (Content Management Servers)
  - Popular ones are:
    - Drupal
    - WordPress
    - Joomla
  - Platforms rich in features and vulnerabilities
  - Open source
  - Some of the vulnerabilities are:
    - SQL Injection
    - Default and weak passwords
    - Errors reveal sensitive information
    - By default Directory listing



# Scanning (cont.)

---

- Attackers maintain a dictionary of vulnerabilities and corresponding exploits.
- For example, if they find an application and its version running on a port. They know whether this version of the application is vulnerable or not. They use their dictionary to verify it.
- Now we understand how to exploit the vulnerabilities.

# Ethical Hacking Methodology





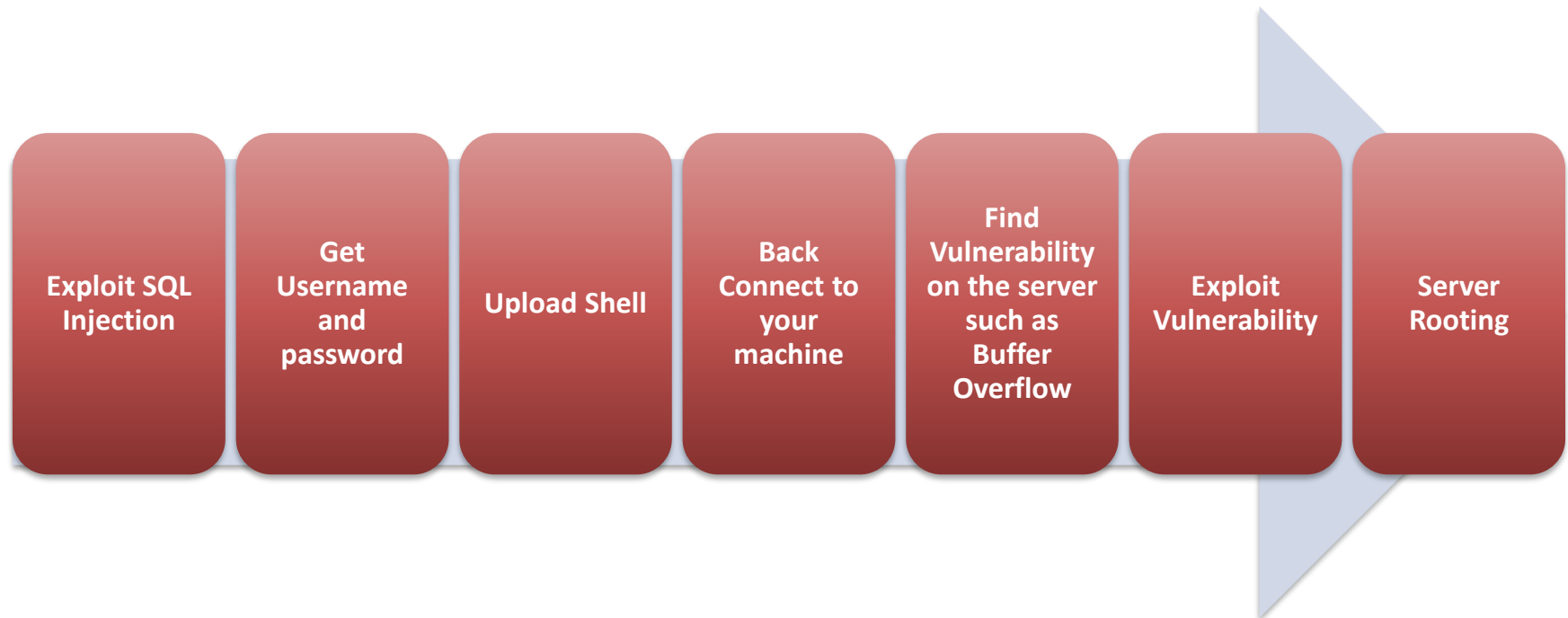
# Attack

- Suppose these are the vulnerabilities we found in the system:
  - SQL Injection - **SQLMap, SQLNinja etc.**
  - Buffer Overflow
- Now we will see how we hack into the system by exploiting these weaknesses.



# Attack (cont.)

## How to plan step-by-step to hack a server?





# Attack (cont.)

- **Exploiting SQL Injection**

- Idea of exploiting SQL injection is to get access to the data and find out what is the admin username and password on the website.
- Once we know this, we can login and upload our shell on the server through which we can escalate our privileges.
- In vulnerability scanning phase, we have identified SQL injection vulnerability in a server. Now we exploit that vulnerability manually and using a automated tool SQL Map.



# Attack (cont.)

- OWASP Web Goat SQL Injection

## General Goal(s):

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

Enter your last name:

```
SELECT * FROM user_data WHERE last_name = 'Your Name'
```

No results matched. Try Again.



# Attack (cont.)

- OWASP Web Goat SQL Injection

## General Goal(s):

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

Enter your last name:

```
SELECT * FROM user_data WHERE last_name = 'Smith'
```

| USERID | FIRST_NAME | LAST_NAME | CC_NUMBER     | CC_TYPE | COOKIE | LOGIN_COUNT |
|--------|------------|-----------|---------------|---------|--------|-------------|
| 102    | John       | Smith     | 2435600002222 | MC      |        | 0           |
| 102    | John       | Smith     | 4352209902222 | AMEX    |        | 0           |





# Attack (cont.)

- OWASP Web Goat SQL Injection

Query

## General Goal(s):

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

**\* Congratulations. You have successfully completed this lesson.**

**\* Now that you have successfully performed an SQL injection, try the same type of attack on a parameterized query. Restart the lesson if you wish to return to the injectable query.**

Enter your last name:

```
SELECT * FROM user_data WHERE last_name = 'Smith' OR '1'='1'
```

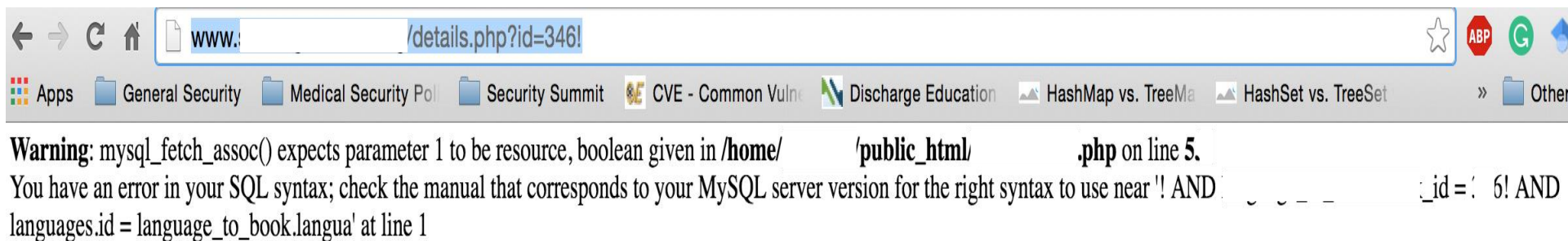
| USERID | FIRST_NAME | LAST_NAME            | CC_NUMBER     | CC_TYPE | COOKIE | LOGIN_COUNT |
|--------|------------|----------------------|---------------|---------|--------|-------------|
| 101    | Joe        | Snow                 | 987654321     | VISA    |        | 0           |
| 101    | Joe        | Snow                 | 2234200065411 | MC      |        | 0           |
| 102    | John       | Smith                | 2435600002222 | MC      |        | 0           |
| 102    | John       | Smith                | 4352209902222 | AMEX    |        | 0           |
| 103    | Jane       | Plane                | 123456789     | MC      |        | 0           |
| 103    | Jane       | Plane                | 333498703333  | AMEX    |        | 0           |
| 10312  | Jolly      | Hershey              | 176896789     | MC      |        | 0           |
| 10312  | Jolly      | Hershey              | 333300003333  | AMEX    |        | 0           |
| 10323  | Grumpy     | youaretheweakestlink | 33413003333   | AMEX    |        | 0           |
| 15603  | Peter      | Sand                 | 123609789     | MC      |        | 0           |
| 15603  | Peter      | Sand                 | 338893453333  | AMEX    |        | 0           |
| 15613  | Joesph     | Something            | 33843453533   | AMEX    |        | 0           |

Admin



# Attack (cont.)

- Identify the SQL Injection by changing the URL parameter.
- Type: inurl .php?id=
- Change the id value. For instance, if id=10, change it to id=10!. See the example below.





# Attack (cont.)

---

- Steps to perform SQL Injection
  - Find vulnerable link (Vulnerability scanning)
  - Find the databases on the vulnerable website
  - Find the relevant tables containing username and passwords
  - Get columns of the table
  - Get data from the table
- SQLMap performs all such actions automatically. You need to provide vulnerable link to it.
- You can also run it as commands on cmd.



# Attack (cont.)

- **SQL MAP Commands**

- **Check if link is vulnerable to SQL Injection**

python sqlmap.py -u <http://www.example.com/authors.php?id=100>

- **Discover Databases**

python sqlmap.py -u [http:// www.example.com/authors.php?id=100](http://www.example.com/authors.php?id=100) --dbs

- **Find Tables of a particular database**

python sqlmap.py -u <http:// www.example.com/authors.php?id=100> --tables  
-D databasename

- **Find the columns**

python sqlmap.py -u <http:// www.example.com/authors.php?id=100> --  
columns -D databasename -T users

- **Get data from the table**

python sqlmap.py -u <http:// www.example.com/authors.php?id=100> --dump  
-D databasename -T users



# Attack (cont.)

- **Modify the Request**

- Suppose after exploiting SQL Injection we have the admin username and password.
- We need to login and upload our shell.
- Waf performs sanitizing that which type of file is being uploaded on the server. So, we need to by-pass the waf.
- We can use Tamper/Scapy to perform this task.
- We can change the format of the shell while uploading and use Tamper browser plug-in to capture the http request to change the file extension to original before it is sent to the server.



# Attack (cont.)

- Source: <http://anonsquad.blogspot.com/2014/02/tutorial-shell-uploading-guide.html>

localhost/sys-options.php

wp\_nav\_menu

Windows-1251

Uname: Windows NT SWASHATA-PC 6.1 build 7601 (Windows 7 Ultimate Edition Service Pack 1) AMD64 [exploit-db.com]  
User: 0 ( SYSTEM ) Group: 0 ( ? )  
Php: 5.3.8 Safe mode: OFF [ phpinfo ] Datetime: 2011-11-29 10:32:46  
Hdd: 48.83 GB Free: 19.24 GB (39%)  
Cwd: D:/wamp/www/ drwxrwxrwx [ home ]  
Drives: [ c ][ d ][ e ][ f ][ g ][ h ][ i ][ j ][ k ][ l ][ m ]

Server IP: 127.0.0.1  
Client IP: 127.0.0.1

[ Sec. Info ] [ Files ] [ Console ] [ Sql ] [ Php ] [ Safe mode ] [ String tools ] [ Bruteforce ] [ Network ] [ Logout ] [ Self remove ]

### File manager

| Name            | Size     | Modify              | Owner/Group | Permissions | Actions |
|-----------------|----------|---------------------|-------------|-------------|---------|
| [ .. ]          | dir      | 2011-10-27 21:10:53 | 0/0         | drwxrwxrwx  | R T     |
| coded.txt       | 30.21 KB | 2011-11-28 16:18:05 | 0/0         | -rw-rw-rw-  | R T E D |
| decoded.txt     | 66.32 KB | 2011-11-28 16:24:37 | 0/0         | -rw-rw-rw-  | R T E D |
| decrypt.php     | 521 B    | 2011-11-28 16:17:02 | 0/0         | -rw-rw-rw-  | R T E D |
| hack.php        | 3.38 KB  | 2011-11-27 08:47:23 | 0/0         | -rw-rw-rw-  | R T E D |
| index.php       | 20.74 KB | 2011-10-27 21:10:29 | 0/0         | -rw-rw-rw-  | R T E D |
| nenens.php      | 3.84 KB  | 2011-11-28 17:03:51 | 0/0         | -rw-rw-rw-  | R T E D |
| script.php      | 1.69 KB  | 2011-11-28 15:45:01 | 0/0         | -rw-rw-rw-  | R T E D |
| source.php      | 28 B     | 2011-11-28 16:24:02 | 0/0         | -rw-rw-rw-  | R T E D |
| sys-options.php | 64.97 KB | 2011-11-28 16:35:36 | 0/0         | -rw-rw-rw-  | R T E D |
| testmysql.php   | 190 B    | 2010-12-31 04:10:06 | 0/0         | -rw-rw-rw-  | R T E D |

Copy >>

Change dir: D:/wamp/www/ >>

Read file: >>

Make dir: (Writeable) >>

Make file: (Writeable) >>

Execute: netstat -n|grep :80|wc -l >>

Upload file: (Writeable) Browse... >>



# Attack (cont.)

- **Back connect**

- Once shell is uploaded, we need to back connect the server so that we can access all the functionality of the uploaded shell and perform privilege escalation attacks.
- For back connect you need to specify the IP address of your computer and Port on which you want to connect on the shell.
- You have to shutdown your firewall and router should be configured for the port forwarding feature.
- **Command for listening:** `nc -v ipaddress port`
- Start listening for the connection on the port specified.



# Attack (cont.)

- **Server Rooting**

- If connect is successful, you should be able to run unix commands such as:
  - ls
  - uname -a
  - whoami
- Download the specific exploit on the server using wget command
- Use chmod 777 exploit for the full permission
- Execute exploit.
- If successful, whoami should say root.





# Attack (cont.)

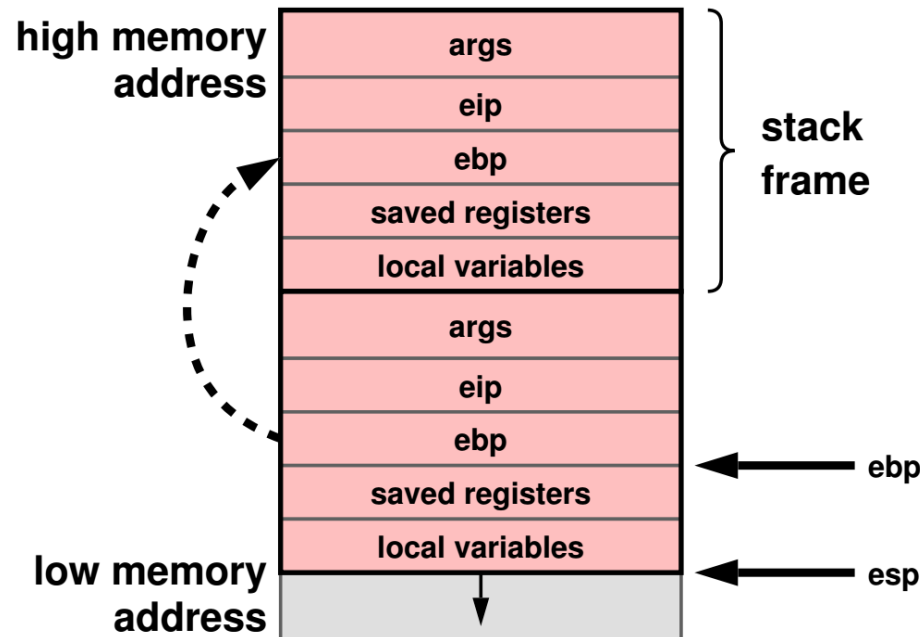
- **Buffer Overflow (BO)**

- It is a vulnerability where a software, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.
- Overwriting values of the IP (Instruction Pointer), BP (Base Pointer) and other registers causes exceptions, segmentation faults etc.
- Consists of overflowing the heap or stack depending on the code that developer has written.



# Attack (cont.)

- Understanding Stack



- Source: Slide of CSCI 402 Operating System: Basic Concepts (Prof. Cheng)



# Attack (cont.)

```
#include <stdio.h>
int main(int argc, char **argv)
{
 char buf[8]; // buffer for eight characters
 gets(buf); // read from stdio (sensitive function!)
 printf("%s\n", buf); // print out data stored in buf
 return 0; // 0 as return value
}
```

```
rezos@spin ~/inzynieria $./bo-simple // program start
1234 // we enter "1234" string from the keyboard
1234 // program prints out the content of the buffer
rezos@spin ~/inzynieria $./bo-simple // start
123456789012 // we enter "123456789012"
123456789012 // content of the buffer "buf" ?!?!
Segmentation fault // information about memory segmentation fault
```

Source: [https://www.owasp.org/index.php/Buffer\\_overflow\\_attack](https://www.owasp.org/index.php/Buffer_overflow_attack)



# Attack (cont.)

- **Exploit the BO Vulnerability (CIJ)**
  - Crash the program by exploiting Buffer overflow.
  - Injecting Malicious code in the program
    - Give the malicious input the program when it crashes.
  - Jumping to the Malicious Code
    - When program runs, your malicious code is inside the memory. You need to overwrite instruction pointer with address so that it jumps to place where malicious code is present.
    - Find the address by understanding the program stack and overwrite the IP.



# Attack (cont.)

- **DDOS Attacks**

- Network bandwidth by flooding
  - DNS Amplification
- Server resources
  - TCP SYN flooding
  - HTTP GET and HTTP POST
  - Slowloris
- Destroy the function of the server or application

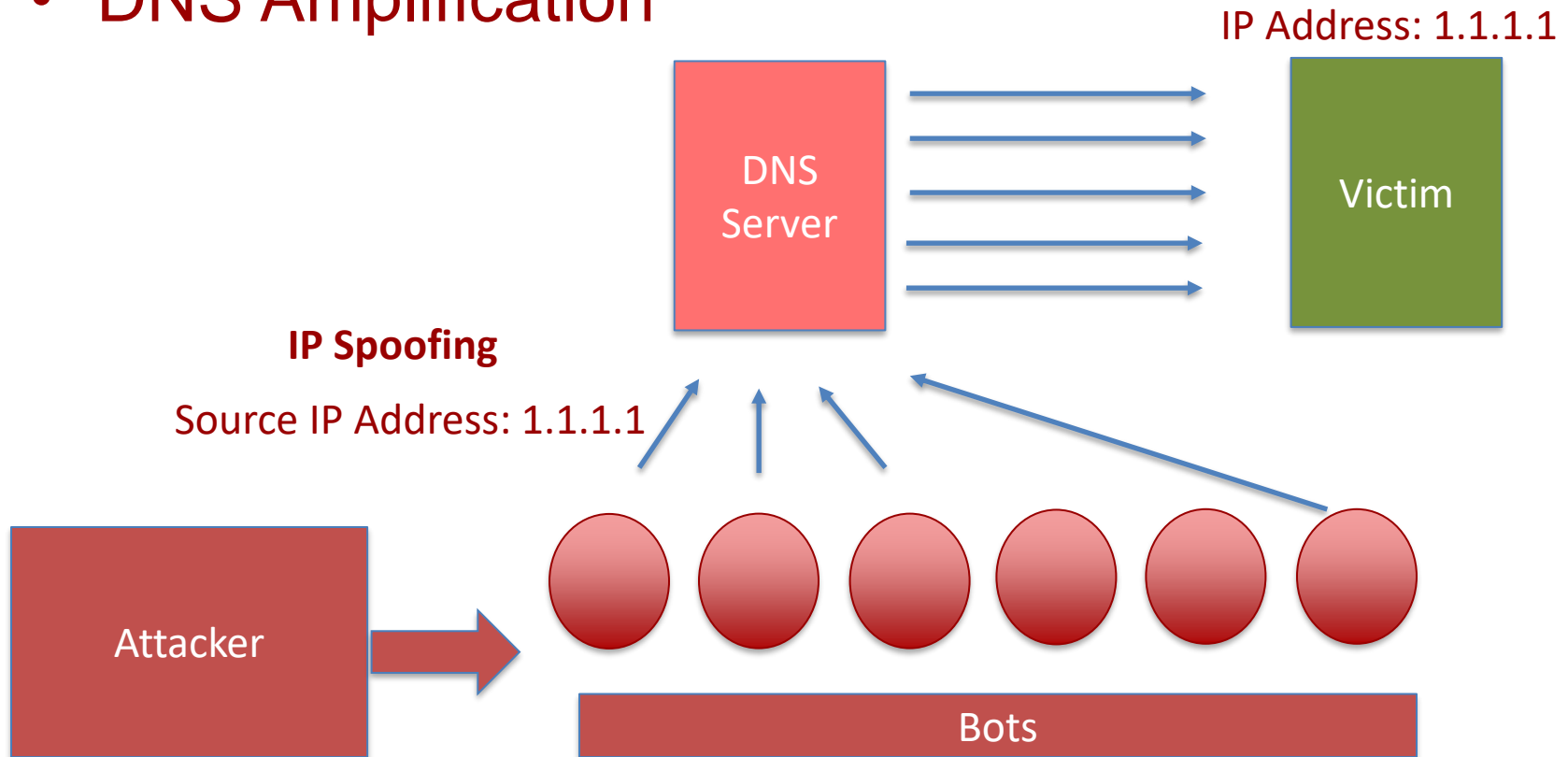
- **Tools:**

- Trinoo
- LOIC (Low Orbit In Cannon)
- TFN2K (Tribe Flood Network)



# Attack (cont.)

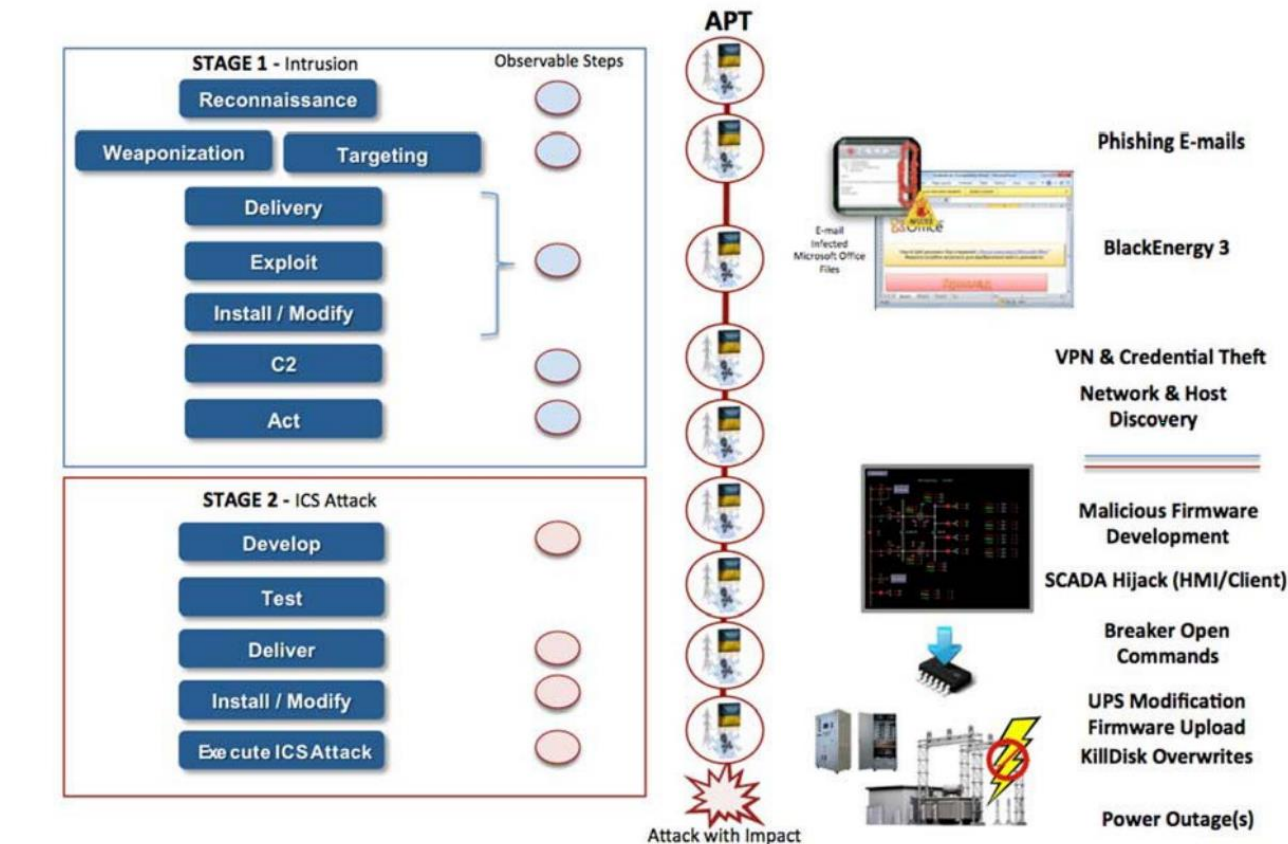
- DNS Amplification



# Case Study: Ukraine Power Grid Attack



- Source: [http://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC\\_SANS\\_Ukraine\\_DUC\\_18Mar2016.pdf](http://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf)





# Tools

|                     |                                                                            |
|---------------------|----------------------------------------------------------------------------|
| <b>Acunetix</b>     | Web Application Security Scanner                                           |
| <b>BurpSuite</b>    | Web Application Security Scanner                                           |
| <b>Veracode</b>     | Application security mobile, web and 3 <sup>rd</sup> party apps.           |
| <b>NMap</b>         | Network Scanning and debugging                                             |
| <b>Wireshark</b>    | Network protocol analyzer for Unix and Windows.                            |
| <b>NeXpose</b>      | Vulnerability Management Software                                          |
| <b>Nessus</b>       | Vulnerability Scanner on Network and applications                          |
| <b>Metasploit</b>   | Penetration Testing tool. Read: Metasploit The penetration guide (reading) |
| <b>FOCA</b>         | Tool to find metadata and hidden information in the documents its scans.   |
| <b>Scapy/Tamper</b> | Packet Generation and Manipulation Program                                 |





# Tools (cont.)

|                         |                                                                  |
|-------------------------|------------------------------------------------------------------|
| <b>Fuzzer</b>           | Manipulating network protocol manipulation                       |
| <b>AirGrab</b>          | Wireless network scanning tool                                   |
| <b>Wi-Fi radar</b>      | Wireless network scanning tool                                   |
| <b>Acrylic Wi-Fi</b>    | Wireless network scanning tool                                   |
| <b>Aircrack-ng</b>      | Wireless network scanning tool                                   |
| <b>Angry IP scanner</b> | Network Scanning                                                 |
| <b>Netcat</b>           | Network Scanning and debugging                                   |
| <b>Nikto2</b>           | Network Scanning and debugging                                   |
| <b>Sulley</b>           | Fuzzing framework for fuzzing files, network protocols CLAs etc. |



# Red Team and Penetration Testing Procedures and Tools

*M.S. Candidate: Matthew Jackoski*

**Lecture 6**  
15 February 2017  
OHE 100C

# Five Stages of Penetration Testing

---



1. Reconnaissance
2. Scanning
3. Gaining Access
4. Maintaining Access
5. Covering Tracks

# Reconnaissance



- Probably the longest stage – lasting weeks to months
- "If I had eight hours to chop down a tree, I'd spend the first six of them sharpening my axe."-Abraham Lincoln
- Learning as much about the system as possible through a variety of different sources:
  - Internet Searches
  - Social Engineering
  - Dumpster Diving
  - Domain Name management/ search services
  - Non-intrusive network scanning
- **MAKE SURE TO DOCUMENT**



# Scanning

- Scanning the perimeter and internal network devices looking for weaknesses.
- Weaknesses include:
  - Open ports
  - Open services
  - Vulnerable Applications (including Operating Systems)
  - Weak protection of data in transit
  - Make and Model of LAN/WAN equipment
- MAKE SURE TO DOCUMENT



# Gaining Access

---

- This is accomplished by using all of information collected in the previous two stages.
  - Exploiting previously discovered vulnerabilities
  - Social Engineering
- Need to avoid detection.
- MAKE SURE TO DOCUMENT



# Maintaining Access

---

- Creating a foothold in the network.
  - Installing a backdoor.
- This action is easier for intrusion detection systems to catch.
- **MAKE SURE TO DOCUMENT**



# Covering Your Tracks

---

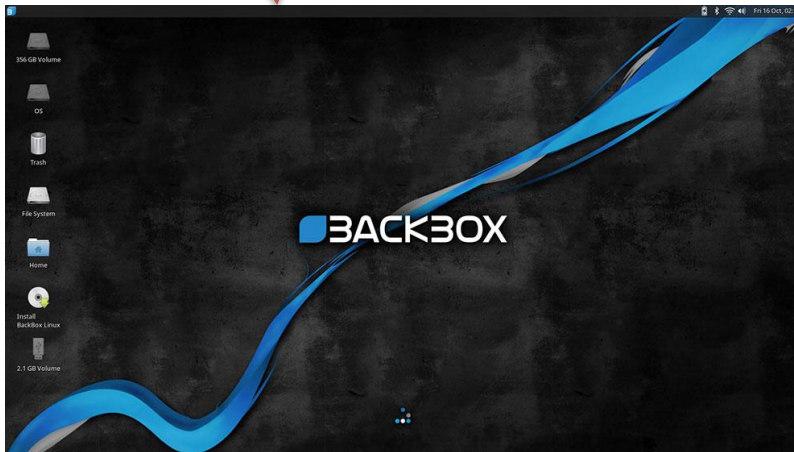
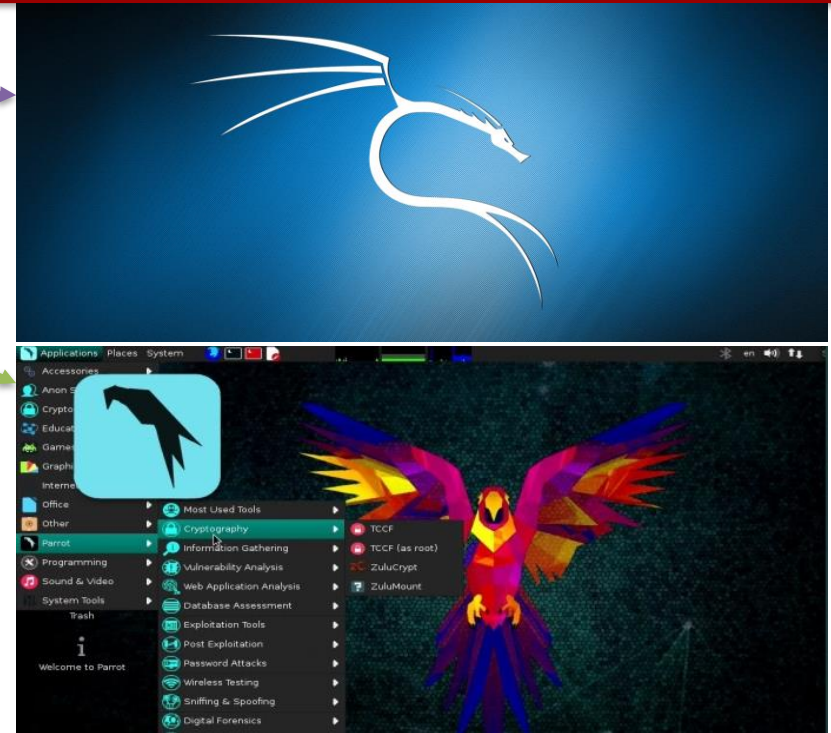
- Need to prove that an attacker can cover his tracks.
- This is mostly accomplished by editing/destroying audit logs.
- **MAKE SURE TO DOCUMENT**



# Security Minded Linux Distributions



- Kali
- Parrot Os
- BackBox



# Most Popular Penetration Testing Tools

---

- Nmap
- Metasploit Framework
- John The Ripper
- THC Hydra
- OWASP Zed
- Wireshark
- Aircrack-ng
- Maltego
- Cain and Abel (Cain)
- Nikto Website Vulnerability Scanner



# Nmap

- Used to discover hosts and services on a computer network.
- Creates a “map” of the network.
- Sends packets to host and analyzes response to generate map.
- Examples of features:
  - Os detection
  - Open Ports
  - Can adapt to latency and congestion



# Metasploit Framework

---

- Tool for executing exploit code against remote machines.
- Basic steps:
  - Choose and configure an exploit
  - Checking whether target is susceptible to the chosen exploit (optional)
  - Choosing and configuring payload
  - Choosing and encoding a stealth method
  - Executing the exploit



# John The Ripper

---

- Password cracking software
- Combines multiple password crackers into one program
- Auto-detects password hash types
- Can be run against various encrypted password formats including:



# THC Hydra

- Password cracker that works in conjunction with John the Ripper
- Fast and stable Network Login Hacking Tool
- Uses brute force or dictionary attacks to login to a webpage
- Supports multiple protocols:
  - Mail
    - POP3
    - IMAP
    - Etc
  - Databases
  - LDAP
  - SMB
  - VNC
  - SSH



# OWASP Zed

- Application Security Scanner
- When used as proxy server, it allows for the user to manipulate all of the traffic that passes through using https.
- Features:
  - Intercepting proxy server
  - Traditional and AJAX Web crawlers
  - Automated scanner
  - Passive scanner
  - Forced browsing
  - Fuzzer
  - WebSocket support
  - Scripting languages
  - Plug-n-Hack support



# Wireshark (TShark)

---

- Open source packet analyzer
- Uses pcap to capture packets
- Allows users to see all traffic, not just the traffic addressed to one of the interface's configured addresses and broadcast/multicast traffic.  
(network interface must support promiscuous mode)





# Aircrack-ng

- Software includes:
  - Detector
  - Packet sniffer
  - WEP and WPA/WPA2-PSK cracker and analysis tool for 802.11 wireless LANs

| Name           | Description                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| aircrack-ng    | Cracks <a href="#">WEP</a> keys using the <a href="#">Fluhrer, Mantin and Shamir attack</a> (FMS) attack, PTW attack, and <a href="#">dictionary attacks</a> , and WPA/WPA2-PSK using dictionary attacks. |
| airdecap-ng    | Decrypts WEP or WPA encrypted capture files with known key.                                                                                                                                               |
| airmon-ng      | Placing different cards in monitor mode.                                                                                                                                                                  |
| aireplay-ng    | Packet injector (Linux, and Windows with <a href="#">CommView</a> drivers).                                                                                                                               |
| airodump-ng    | <a href="#">Packet sniffer</a> : Places air traffic into <a href="#">pcap</a> or IVS files and shows information about networks.                                                                          |
| airtun-ng      | Virtual tunnel interface creator.                                                                                                                                                                         |
| packetforge-ng | Create encrypted packets for injection.                                                                                                                                                                   |
| ivstools       | Tools to merge and convert.                                                                                                                                                                               |
| airbase-ng     | Incorporates techniques for attacking client, as opposed to Access Points.                                                                                                                                |
| airdecloak-ng  | Removes WEP cloaking from pcap files.                                                                                                                                                                     |
| airolib-ng     | Stores and manages ESSID and password lists and compute Pairwise Master Keys.                                                                                                                             |
| airserv-ng     | Allows to access the wireless card from other computers.                                                                                                                                                  |
| buddy-ng       | The helper server for easside-ng, run on a remote computer.                                                                                                                                               |
| easside-ng     | A tool for communicating to an access point, without the WEP key.                                                                                                                                         |
| tkiptun-ng     | WPA/TKIP attack.                                                                                                                                                                                          |
| wesside-ng     | Automatic tool for recovering wep key.                                                                                                                                                                    |



# Maltego

- Used for open-source intelligence and forensics
- Allows for creating custom entities, which allows for representation of any type of data.
- Main focus:
  - To analyze real-world relationships between people, groups, websites, domains, networks, internet infrastructure, and affiliations with online services.



# Cain and Abel

- Password recovery tool for Microsoft Windows
- Password cracks are done by dictionary attacks, brute force, and Cryptanalysis
- Features:
  - WEP cracking
  - Speeding up packet capture speed
  - Record VoIP conversations
  - Decoding scrambled passwords
  - Calculating hashes
  - Traceroute
  - Revealing password boxes
  - Uncovering cached passwords
  - Dumping protected storage passwords
  - ARP spoofing
  - IP to MAC Address resolver
  - Network Password Sniffer
  - LSA secret dumper



# Nikto Website Scanner

---

- Open Source web server scanner.
- Performs comprehensive tests against web servers, including over 6700 potentially dangerous files/CGIs.
- Checks for outdated versions
- Checks for specific problems associated with over 270 servers.
- Checks server configurations for items such as the presence of multiple index files, HTTP server options, and attempts to identify installed web servers and software